ARMY RESEARCH LABORATORY

Image Processing REST Web Services

by Robert P. Winkler and Chris Schlesiger

ARL-TR-6393 **March 2013**

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory
Adelphi, MD 20783-1197

ARL-TR-6393 **March 2013**

Image Processing REST Web Services

Robert P. Winkler and Chris Schlesiger
Computational and Information Sciences Directorate, ARL

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED (From - To)
March 2013	Final	

4. TITLE AND SUBTITLE
Image Processing REST Web Services

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Robert P. Winkler and Chris Schlesiger

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
ATTN: RDRL-CII-B
2800 Powder Mill Road
Adelphi, MD 20783-1197

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-6393

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

In this report, we present a readily extensible architecture for image processing based on representational state transfer (REST) Web services. Relying on just two types of resources, images and collections of images, any number of new image processing algorithms can be readily added. Three image processing algorithms developed by the U.S. Army Research Laboratory's Sensors and Electronic Devices Directorate are integrated into the architecture to demonstrate extensibility.

15. SUBJECT TERMS
REST Web Services, image processing, deblurring, contrast-enhancement, super resolution

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Robert P. Winkler
Unclassified	Unclassified	Unclassified	UU	66	19b. TELEPHONE NUMBER (*Include area code*) (301) 394-1689

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

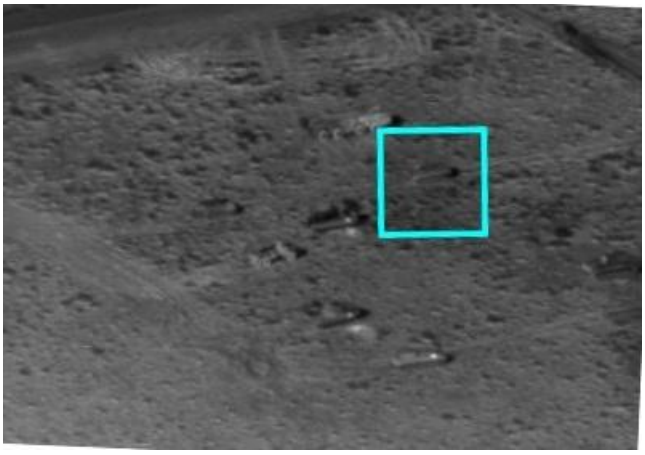
List of Images

List of Listings

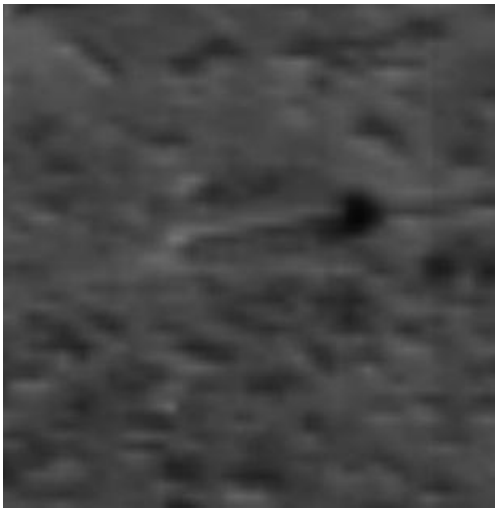
1. Introduction

This report describes the Web services exposed via representational state transfer (REST) for client applications to access a suite of image processing algorithms developed by the U.S. Army Research Laboratory's (ARL) Sensors and Electronic Devices Directorate (SEDD). REST is a software architecture used by the World Wide Web and introduced by Roy Fielding in his doctoral dissertation (Fielding 2000). One distinguishing characteristic of the architecture is that communications between the client and the server are stateless. Requests are initiated by the client, the server processes the request, returns a response, and the entire transaction is considered atomic. This report provides the interface definitions for three image processing algorithms developed by SEDD: deblurring (Young, Driggers and Teaney, et al. 2007), contrast enhancement (Maschel and Young 2012), and super resolution (Young and Driggers 2006). These algorithms are often chained together to produce crisper images. An example is shown in image 1.

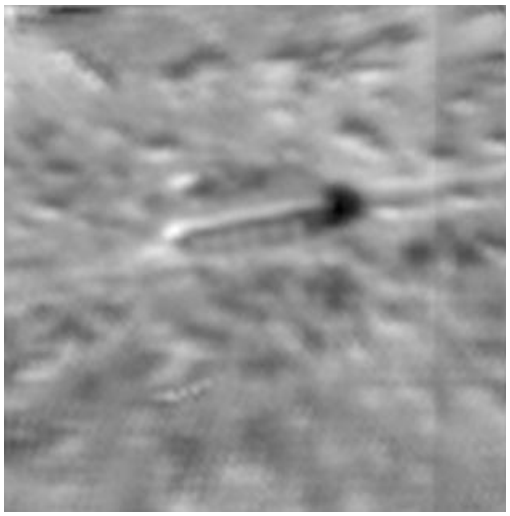
First, we discuss the resources associated with these Web services and then we describe the Web services associated with images, collections, deblurring, contrast enhancement, and super resolution.



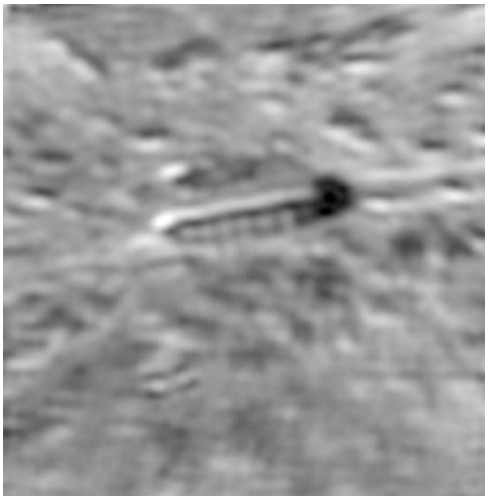
1. Original Image with Target Chip to Super Resolve



2. Unenhanced extracted target chip



3. Super-resolved target chip



4. Super-resolved, deblurred target chip

5. Super-resolved, deblurred and contrast-enhanced target chip

Image 1. Chaining the image processing algorithms.

2. Resources

There are two types of resources associated with these services: images and collections. Uploaded images can be any mime type supported by the Windows .NET framework (bmp, emf, exif, gif, icon, jpeg, png, tiff, or wmf). The image processing algorithms currently only work on 32-bit grayscale images, so images are converted to 32-bit grayscale before being submitted to the algorithms. Images can also be annotated with text overlaid on the image and/or pen strokes, which conform to the W3C InkML recommendation (W3C 2011). Collections can be collections of images or of other collections. Every resource is associated with a universal resource indicator (URI). Images and collections and their metadata are retrieved with GET, created with POST, and destroyed with DELETE. Image and collection metadata can also be replaced with PUT. When an image or collection is added, the HTTP response body contains the server generated URI. When an image is retrieved via GET, the base 64 ASCII encoding of the image binary data is contained in the response body along with any associated metadata. The Extensible Markup Language (XML) schema definition for the image metadata is shown in listing 1 and for the collection metadata in listing 2.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="ARLImageProcessingMetadata"
targetNamespace="http://arl.army.milhttp://{host}:{port}/ARLImageProcessing/Metadata.xsd"
elementFormDefault="qualified"
xmlns="http://arl.army.milhttp://{host}:{port}/ARLImageProcessing/Metadata.xsd"
xmlns:mstns="http://arl.army.milhttp://{host}:{port}/ARLImageProcessing/Metadata.xsd"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:inkml="http://www.w3.org/2003/InkML">
<xs:import id="inkml" namespace="http://www.w3.org/2003/InkML"/>
<xs:element name="Metadata">
  <xs:complexType>
         <xs:sequence>
          <xs:element name="MediaType" type="xs:string" minOccurs="0"/>
          <xs:element name="Title" type="xs:string" minOccurs="0"/>
          <xs:element name="DateCreated" type="xs:dateTime" minOccurs="0"/>
          <xs:element name="ParentURI" type="xs:string" minOccurs="0"/>
          <xs:element name="Chip" minOccurs="0">
           <xs:complexType>
            <xs:attribute name="UpperLeftX" type="xs:unsignedInt" use="required"/>
            <xs:attribute name="UpperLeftY" type="xs:unsignedInt" use="required"/>
            <xs:attribute name="LowerRightX" type="xs:unsignedInt" use="required"/>
            <xs:attribute name="LowerRightY" type="xs:unsignedInt" use="required"/>
           </xs:complexType>
          </xs:element>
          <xs:element name="ContrastEnhancementParameters" minOccurs="0">
           <xs:complexType>
            <xs:attribute name="PercentageLow" type="xs:double" use="required"/>
            <xs:attribute name="PercentageHigh" type="xs:double" use="required"/>
           </xs:complexType>
          </xs:element>
          <xs:element name="DeblurParameters" minOccurs="0">
           <xs:complexType>
            <xs:attribute name="Alpha" type="xs:double" use="required"/>
            <xs:attribute name="MG" type="xs:unsignedInt" use="required"/>
           </xs:complexType>
          </xs:element>
          <xs:element name="SuperResolutionParameters" minOccurs="0">
           <xs:complexType>
            <xs:attribute name="Align" type="xs:boolean" use="required"/>
            <xs:attribute name="Inset" type="xs:unsignedInt" use="required"/>
            <xs:attribute name="Quality" type="xs:unsignedInt" use="required"/>
            <xs:attribute name="Speed" type="xs:double" use="required"/>
           </xs:complexType>
          </xs:element>
          <xs:element name="Annotations" minOccurs="0">
           <xs:complexType>
            <xs:sequence>
             <xs:element name="Text" minOccurs="0" maxOccurs="unbounded">
              <xs:complexType>
               <xs:attribute name="X" type="xs:unsignedInt" use="required"/>
               <xs:attribute name="Y" type="xs:unsignedInt" use="required"/>
               <xs:attribute name="Font" type="xs:string"/>
               <xs:attribute name="Size" type="xs:string"/>
               <xs:attribute name="Color" type="xs:string"/>
              </xs:complexType>
             </xs:element>
             <xs:element name="ink" type="inkml:ink.type" minOccurs="0"/>
            </xs:sequence>
          </xs:complexType>
          </xs:element>
          <xs:element name="Media" type="xs:base64Binary" minOccurs="0"/>
         </xs:sequence>
         <xs:attribute name="URI" type="xs:string"/>
  </xs:complexType>
</xs:element>
</xs:schema>
```

Listing 1. Image metadata.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="Collection"
  targetNamespace="http://arl.army.milhttp://{host}:{port}/ARLImageProcessing/Collection.xsd"
  elementFormDefault="qualified"
  xmlns="http://arl.army.milhttp://{host}:{port}/ARLImageProcessing/Collection.xsd"
  xmlns:mstns="http://arl.army.milhttp://{host}:{port}/ARLImageProcessing/Collection.xsd"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Collection">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Title" type="xs:string" minOccurs="0"/>
        <xs:element name="DateCreated" type="xs:dateTime" minOccurs="0"/>
        <xs:element name="ImageURI" type="xs:string" minOccurs="1" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="URI" type="xs:string"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Listing 2. Collection metadata.

3. Image Services

In this section, we describe the Web services available for raw images.

3.1 PostImage

Upload an image and return its server-generated URI. The data flow for PostImage is shown in figure 1.

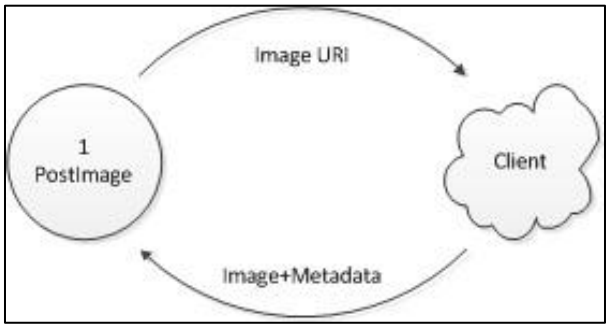


Figure 1. PostImage.

Request Syntax:
POST http://{host}:{port}/ARLImageProcessing/images

Input Parameters	Definition	Use
MediaType	the mime type of the image	required
Title	the title of the image	optional
ParentURI	the URI of the parent image	optional
Annotations	textual and graphical annotations	0..1
Annotations.Text	any textual annotations for the image	optional
Annotations.Text.X	X pixel position for the start of the text	required
Annotations.Text.Y	Y pixel position for the start of the text	required
Annotations.Text.Font	font for the text	optional
Annotations.Text.Size	size of the text	optional
Annotations.Text.Color	ARGB color for the text	optional
Annotations.ink	any InkML markup for the image	optional
Media	the base 64 ASCII encoding of the image	required

Output Parameters	Definition	Cardinality
string	the server-generated URI of the image	1..1

```
POST http://mule01.lsn.arl:8080/ARLImageProcessing/images HTTP/1.1


<?xml version="1.0" encoding="utf-8" ?>
<Metadata URI="http://mule01.lsn.arl:8080/ARLImageProcessing/images/49e667c6-d309-4002-8e9c-
4d6b1d98deaa"
  xmlns="http://arl.army.milhttp://{host}:{port}/ARLImageProcessing/Metadata.xsd"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <MediaType>image/bmp</MediaType>
  <Title>Exit sign frame 13</Title>
  <ParentURI>http://mule01.lsn.arl:8080http://{host}:{port}/ARLImageProcessing/images/0ab3123c-15f0-
4797-9ee2-7fc8625bd511</ParentURI>
  <Annotations>
    <Text X="191" Y="186" Font="Segoe360" Size="12" Color="#FFFF0000">Some Text</Text>
    <Text X="227" Y="315" Font="Segoe360" Size="12" Color="#FFFF0000">Some More Text</Text>
    <ink xmlns="http://www.w3.org/2003/InkML">
      <definitions>
        <brush xml:id="1">
          <brushProperty name="width" value="2.00314960629921" />
          <brushProperty name="height" value="2.00314960629921" />
          <brushProperty name="color" value="#FF00FFFF" />
          <brushProperty name="tip" value="ellipse" />
        </brush>
      </definitions>
      <trace brushRef="1">38.5566032170582 192.222980271295, 39.4092523017537 191.65161538904,
39.4069967680897 190.702712261337</trace>
    </ink>
  </Annotations>
<Media>Qk028AAAAAAA ...</Media>
</Metadata>
```

Listing 3. Example PostImage request.

```
HTTP/1.1 201 Created

<?xml version="1.0" encoding="utf-8"?>
<string>http://mule01.lsn.arl:8080http://{host}:{port}/ARLImageProcessing/images/eceea312-2138-4038-933e-359f477f2460</string>
```

Listing 4. Example PostImage response.

3.2 GetImage

Retrieve an image and its metadata. The data flow for GetImage is shown in figure 2.

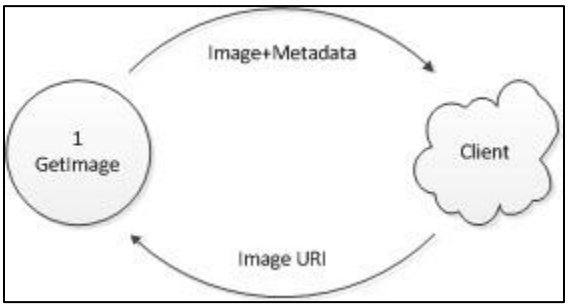


Figure 2. GetImage.

Request Syntax:
GET http://{host}:{port}/ARLImageProcessing/images/{imageID}

Input Parameters	Definition	Use
imageID	the guid of image to retrieve	required

Output Parameters	Definition	Cardinality
URI	the URI of the image	1..1
MediaType	the mime type of the image	1..1
Title	the title of the image	0..1
DateCreated	sever-generated timestamp of when the image was uploaded	1..1
ParentURI	the URI of the parent image	0..1
Annotations	textual and graphical annotations	0..1
Annotations.Text	any textual annotations for the image	0..1
Annotations.Text.X	X pixel position for the start of the text	1..1
Annotations.Text.Y	Y pixel position for the start of the text	1..1
Annotations.Text.Font	font for the text	0..1
Annotations.Text.Size	size of the text	0..1
Annotations.Text.Color	ARGB color for the text	0..1
Annotations.ink	any InkML markup for the image	0..1
Media	the base 64 ASCII encoding of the image	1..1

```
GET http://{host}:{port}/ARLImageProcessing/images/49e667c6-d309-4002-8e9c-4d6b1d98deaa HTTP/1.1
```

Listing 5. Example GetImage request.

```
HTTP/1.1 200 OK

<?xml version="1.0" encoding="utf-8" ?>
<Metadata URI="http://mule01.lsn.arl:8080http://{host}:{port}/ARLImageProcessing/images/49e667c6-d309-
4002-8e9c-4d6b1d98deaa">
   <MediaType>image/bmp</MediaType>
   <Title>Exit sign frame 13</Title>
   <DateCreated>2012-05-24T15:31:55.8375601-04:00</DateCreated>
   <ParentURI>http://mule01.lsn.arl:8080http://{host}:{port}/ARLImageProcessing/images/0ab3123c-15f0-4797-
9ee2-7fc8625bd511</ParentURI>
   <Annotations>
      <Text X="191" Y="186" Font="Segoe360" Size="12" Color="#FFFF0000">Some Text</Text>
      <Text X="227" Y="315" Font="Segoe360" Size="12" Color="#FFFF0000">Some More Text</Text>
      <ink>
         <definitions>
            <brush xml:id="1">
               <brushProperty name="width" value="2.00314960629921" />
               <brushProperty name="height" value="2.00314960629921" />
               <brushProperty name="color" value="#FF00FFFF" />
               <brushProperty name="tip" value="ellipse" />
            </brush>
         </definitions>
         <trace brushRef="1">38.5566032170582 192.222980271295, 39.4092523017537 191.65161538904,
39.4069967680897 190.702712261337</trace>
      </ink>
   </Annotations>
   <Media>Qk028AAAAAAA ...</Media>
</Metadata>
```

Listing 6. Example GetImage response.

3.3 GetImageMetadata

Retrieve an image's metadata. The data flow for GetImageMetadata is shown in figure 3.

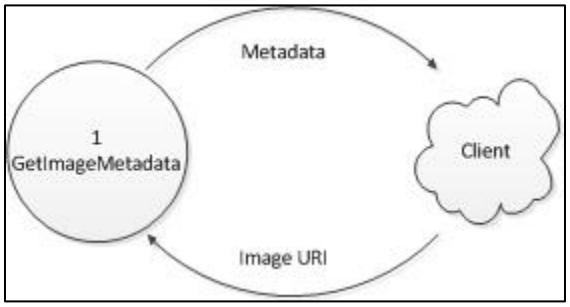


Figure 3. GetImageMetadata.

Request Syntax:
GET http://{host}:{port}/ARLImageProcessing/images/{imageID}/metadata

Input Parameters	Definition	Use
imageID	the guid of image to retrieve	required

Output Parameters	Definition	Cardinality
URI	the URI of the image	1..1
MediaType	the mime type of the image	1..1
Title	the title of the image	0..1
ParentURI	the URI of the parent image	0..1
Annotations	textual and graphical annotations	0..1
Annotations.Text	any textual annotations for the image	0..1
Annotations.Text.X	X pixel position for the start of the text	1..1
Annotations.Text.Y	Y pixel position for the start of the text	1..1
Annotations.Text.Font	font for the text	0..1
Annotations.Text.Size	size of the text	0..1
Annotations.Text.Color	ARGB color for the text	0..1
Annotations.ink	any InkML markup for the image	0..1
Media	the base 64 ASCII encoding of the image	1..1

```
GET http://{host}:{port}/ARLImageProcessing/images/49e667c6-d309-4002-8e9c-4d6b1d98deaa HTTP/1.1
```

Listing 7. Example GetImageMetadata request.

```
HTTP/1.1 200 OK

<?xml version="1.0" encoding="utf-8" ?>
<Metadata URI="http://mule01.lsn.arl:8080http://{host}:{port}/ARLImageProcessing/images/49e667c6-d309-
4002-8e9c-4d6b1d98deaa">
   <MediaType>image/bmp</MediaType>
   <Title>Exit sign frame 13</Title>
   <DateCreated>2012-05-24T15:31:55.8375601-04:00</DateCreated>
   <ParentURI>http://mule01.lsn.arl:8080http://{host}:{port}/ARLImageProcessing/images/0ab3123c-15f0-4797-
9ee2-7fc8625bd511</ParentURI>
   <Annotations>
      <Text X="191" Y="186" Font="Segoe360" Size="12" Color="#FFFF0000">Some Text</Text>
      <Text X="227" Y="315" Font="Segoe360" Size="12" Color="#FFFF0000">Some More Text</Text>
      <ink>
         <definitions>
            <brush xml:id="1">
               <brushProperty name="width" value="2.00314960629921" />
               <brushProperty name="height" value="2.00314960629921" />
               <brushProperty name="color" value="#FF00FFFF" />
               <brushProperty name="tip" value="ellipse" />
            </brush>
         </definitions>
         <trace brushRef="1">38.5566032170582 192.222980271295, 39.4092523017537 191.65161538904,
39.4069967680897 190.702712261337</trace>
      </ink>
   </Annotations>
</Metadata>
```

Listing 8. Example GetImageMetadata response.

3.4 GetImageOverlay

Retrieve a transparent image consisting of just the textual and graphical annotations of an image. Applications that do not want to render the annotations and markup themselves can call this service to retrieve a transparent image of just the markup for an image and then overlay it on top of the image to see the annotations. The data flow for GetImageOverlay is shown in figure 4.

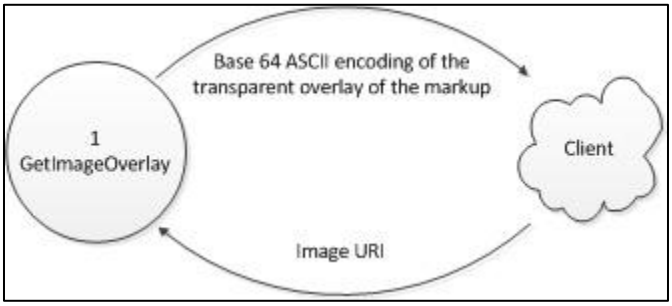


Figure 4. Get ImageOverlay.

Request Syntax:
GET http://{host}:{port}/ARLImageProcessing/images/{imageID}/overlay

Input Parameters	Definition	Use
imageID	the guid of the annotated image	required

Output Parameters	Definition	Cardinality
base64Binary	the base 64 ASCII encoding of the overlay image	1..1

```
GET http://{host}:{port}/ARLImageProcessing/images/9e635628-b064-4240-8dc0-c4ec1750f874/overlay
HTTP/1.1
```

Listing 9. Example GetImageOverlay request.

```
HTTP/1.1 200 OK

<?xml version="1.0" encoding="utf-8" ?>
<base64Binary>iVBORw0KGgoAAAANSUhE ...</base64Binary>
```

Listing 10. Example GetImageOverlay response.

3.5 GetImageList

Retrieve the metadata associated with all of the images including all of the base images as well as those which have been deblurred, contrast-enhanced, or super-resolved. The data flow for GetImageList is shown in figure 5.

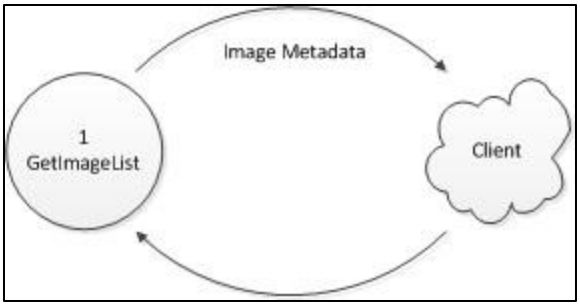


Figure 5. GetImageList.

Request Syntax:
GET http://{host}:{port}/ARLImageProcessing/images

Output Parameters	Definition	Context	Cardinality
Metadata	the metadata associated with a single image	all	0..N
Metadata.URI	the server-generated URI of the image	all	1..1
Metadata.MediaType	the mime type of the image	all	1..1
Metadata.Title	the title of the image	all	0..1
Metadata.DateCreated	sever-generated timestamp of when the image was uploaded	all	1..1
Metadata.ParentURI	the URI of the parent image or collection	all	0..1
Metadata.ContrastEnhancementParameters	parameters used for contrast enhancement	contrast-enhanced	1..1
Metadata.ContrastEnhancementParameters .PercentageLow	lower tail of the accumulated histogram	contrast-enhanced	1..1
Metadata.ContrastEnhancementParameters. PercentageHigh	upper tail of the accumulated histogram	contrast-enhanced	1..1
Metadata.DeblurParameters	the parameters used for this deblurring	deblurred	1..1
Metadata.DeblurParameters.Alpha	the ratio of the standard deviation of the Gaussian filter to the maximum polar frequency	deblurred	1..1
Metadata.DeblurParameters.MG	the magnitude of the inverse Gaussian	deblurred	1..1
Metadata.Chip	the coordinates the chip extracted from the parent images for super resolution	super-resolved	1..1
Metadata.Chip.LowerRightX	the X pixel coordinate of the lower right corner of the chip extracted from the parent images	super-resolved	1..1
Metadata.Chip.LowerRightY	the Y pixel coordinate of the lower right corner of the chip extracted from the parent images	super-resolved	1..1
Metadata.Chip.UpperLeftX	the X pixel coordinate of the upper left corner of the chip extracted from the parent images	super-resolved	1..1

Output Parameters	Definition	Context	Cardinality
Metadata.Chip.UpperLeftY	the Y pixel coordinate of the upper left corner of the chip extracted from the parent images	super-resolved	1..1
Metadata.SuperResolutionParameters	the parameters used for super resolution	super-resolved	1..1
Metadata.SuperResolutionParameters .Align	should the base images be aligned	super-resolved	1..1
Metadata.SuperResolutionParameters .Inset	defines a sub-region with the Chip for frame registration purposes	super-resolved	1..1
Metadata.SuperResolutionParameters .Quality	upsampling factor the reconstruction grid	super-resolved	1..1
Metadata.SuperResolutionParameters .Speed	frequency bandwidth of the super-resolved image	super-resolved	1..1
Metadata.Annotations	textual and graphical annotations	all	0..1
Metadata.Annotations.Text	any textual annotations for the image	all	0..N
Metadata.Annotations.Text.X	X pixel position for the start of the text	all	1..1
Metadata.Annotations.Text.Y	Y pixel position for the start of the text	all	1..1
Metadata.Annotations.Text.Font	optional font the text	all	0..1
Metadata.Annotations.Text.Size	optional size for the font	all	0..1
Metadata.Annotations.Text.Color	optional ARGB color for the text	all	0..1
Metadata.Annotations.ink	any InkML markup for the image	all	0..1

```
GET http://{host}:{port}/ARLImageProcessing/images
```

Listing 11. Example GetImageList request.

```
HTTP/1.1 200 OK

<?xml version="1.0" encoding="utf-8" ?>
<ArrayOfMetadata>
  <Metadata URI="http://mule01.lsn.arl:8080http://{host}:{port}/ARLImageProcessing/images/deblurred/0476ef6a-bbb4-41bd-bbd5-f8117a470514">
    <MediaType>image/png</MediaType>
    <Title>Deblurred Face</Title>
    <DateCreated>2012-08-29T14:15:57.2311694-04:00</DateCreated>
    <ParentURI>http://mule01.lsn.arl:8080http://{host}:{port}/ARLImageProcessing/images/9e635628-b064-4240-8dc0-c4ec1750f874</ParentURI>
    <DeblurParameters Alpha="0.11" MG="9"/>
  </Metadata>
  <Metadata URI="http://mule01.lsn.arl:8080http://{host}:{port}/ARLImageProcessing/images/contrastenhanced/0476ef6a-bbb4-41bd-bbd5-f8117a470514">
    <MediaType>image/png</MediaType>
    <Title>Contrast Enhanced Face</Title>
    <DateCreated>2012-08-29T14:15:57.2311694-04:00</DateCreated>
    <ParentURI>http://mule01.lsn.arl:8080http://{host}:{port}/ARLImageProcessing/images/9e635628-b064-4240-8dc0-c4ec1750f874</ParentURI>
    <ContrastEnhancementParameters PercentageLow="0.5" PercentageHigh="0.002" />
  </Metadata>
  <Metadata URI="http://mule01.lsn.arl:8080http://{host}:{port}/ARLImageProcessing/collections/superresolved/07659a20-6ac3-42bf-a6a4-93dc26527a39">
    <MediaType>image/png</MediaType>
    <Title>Super Resolved Face</Title>
    <DateCreated>2012-08-29T14:18:45.9298657-04:00</DateCreated>
    <ParentURI>http://mule01.lsn.arl:8080http://{host}:{port}/ARLImageProcessing/collections/9618d68b-0f46-4f28-8ef8-7dd4765d64ad</ParentURI>
    <Chip LowerRightX="252" LowerRightY="376" UpperLeftX="152" UpperLeftY="147" />
    <SuperResolutionParameters Align="true" Inset="5" Quality="10" Speed="0.55" />
  </Metadata>
  <Metadata URI="http://mule01.lsn.arl:8080http://{host}:{port}/ARLImageProcessing/images/49e667c6-d309-4002-8e9c-4d6b1d98deaa">
    <MediaType>image/bmp</MediaType>
    <Title>Exit sign frame 13</Title>
    <DateCreated>2012-05-24T15:31:55.8375601-04:00</DateCreated>
    <ParentURI>http://mule01.lsn.arl:8080http://{host}:{port}/ARLImageProcessing/images/0ab3123c-15f0-4797-9ee2-7fc8625bd511</ParentURI>
    <Annotations>
      <Text X="191" Y="186" Font="Segoe360" Size="12" Color="#FFFF0000">Some Text</Text>
      <Text X="227" Y="315" Font="Segoe360" Size="12" Color="#FFFF0000">Some More Text</Text>
      <ink xmlns="http://www.w3.org/2003/InkML">
        <definitions>
          <brush xml:id="1">
            <brushProperty name="width" value="2.00314960629921" />
            <brushProperty name="height" value="2.00314960629921" />
            <brushProperty name="color" value="#FF00FFFF" />
            <brushProperty name="tip" value="ellipse" />
          </brush>
        </definitions>
        <trace brushRef="1">38.5566032170582 192.222980271295, 39.4092523017537 191.65161538904, 39.4069967680897 190.702712261337</trace>
      </ink>
    </Annotations>
  </Metadata>
</ArrayOfMetadata>
```

Listing 12. Example GetImageList response.

3.6 DeleteImage

Delete an image. Note that any references to the image in any collections will be left dangling. The data flow for DeleteImage is shown in figure 6.

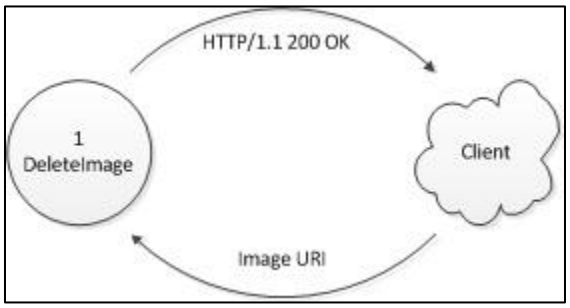


Figure 6. DeleteImage.

Request Syntax:
DELETE http://{host}:{port}/ARLImageProcessing/images/{imageID}

Input Parameters	Definition	Use
imageID	the guid of the image to delete	required

```
DELETE http://{host}:{port}/ARLImageProcessing/images/c734551e-bc64-453b-a361-4dbe17f1f356 HTTP/1.1
```

Listing 13. Example DeleteImage request.

```
HTTP/1.1 200 OK
```

Listing 14. Example DeleteImage response.

3.7 PutImageMetadata

Replace an image's metadata. Note that the image's metadata is entirely replaced and not merged or otherwise checked in any manner. Even the image's URI can be changed or deleted with this method, resulting in not only a corrupt image, but also corrupting any collections it belongs to. To add to the metadata, the current metadata should first be retrieved and copied into the new metadata. This method is typically used to add textual annotations and ink markup to an existing image. The data flow for PutImageMetadata is shown in figure 7.

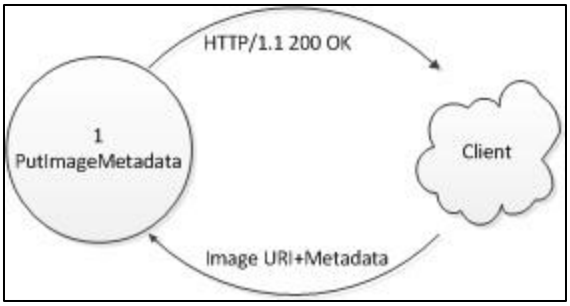


Figure 7. PutImageMetadata.

Request Syntax:
PUT http://{host}:{port}/ARLImageProcessing/images/{imageID}

Input Parameters	Definition	Use
imageID	the guid of image's metadata to replace	required
URI	the URI of the image	copy from current metadata
MediaType	the mime type of the image	copy from current metadata
Title	the title of the image	optional
DateCreated	sever-generated timestamp of when the image was uploaded	copy from current metadata to preserve the original creation date or replace with current date/time to preserve the modification timestamp
ParentURI	the URI of the parent image	copy from current metadata
Annotations	textual and graphical annotations	optional
Annotations.Text	any textual annotations for the image	optional
Annotations.Text.X	X pixel position for the start of the text	required
Annotations.Text.Y	Y pixel position for the start of the text	required
Annotations.Text.Font	font for the text	optional
Annotations.Text.Size	size of the text	optional
Annotations.Text.Color	ARGB color for the text	optional
Annotations.ink	any InkML markup for the image	optional

```
PUT http://{host}:{port}/ARLImageProcessing/images/07659a20-6ac3-42bf-a6a4-93dc26527a39 HTTP/1.1

<?xml version="1.0" encoding="utf-8" ?>
<Metadata URI="http://mule01.lsn.arl:8080http://{host}:{port}/ARLImageProcessing/images/07659a20-6ac3-42bf-a6a4-93dc26527a39">
  <MediaType>image/png</MediaType>
  <Title>Face</Title>
  <DateCreated>2012-08-29T14:18:45.9298657-04:00</DateCreated>
  <ParentURI>http://mule01.lsn.arl:8080http://{host}:{port}/ARLImageProcessing/images/9618d68b-0f46-4f28-8ef8-7dd4765d64ad</ParentURI>
  <Annotations>
    <Text X="191" Y="186" Font="Segoe360" Size="12" Color="#FFFF0000">Some Text</Text>
    <Text X="227" Y="315" Font="Segoe360" Size="12" Color="#FFFF0000">Some More Text</Text>
    <ink>
      <definitions>
        <brush xml:id="1">
          <brushProperty name="width" value="2.00314960629921" />
          <brushProperty name="height" value="2.00314960629921" />
          <brushProperty name="color" value="#FF00FFFF" />
          <brushProperty name="tip" value="ellipse" />
        </brush>
      </definitions>
      <trace brushRef="1">38.5566032170582 192.222980271295, 39.4092523017537 191.65161538904,
39.4069967680897 190.702712261337</trace>
    </ink>
  </Annotations>
</Metadata>
```

Listing 15. Example PutImageMetadata request.

```
HTTP/1.1 200 OK
```

Listing 16. Example PutImageMetadata response.

4. Collection Services

In this section, we describe the Web services associated with collections.

4.1 PostCollection

Upload a collection and return the server-generated URI of the new collection. The typical steps in creating a collection are to first call PostImage to upload number of images while preserving their server-generated URIs. Then call PostCollection with the collection metadata containing the image URIs. Note that collections may also contain other collections so the ImageURI parameter could refer either to an image or another collection. A typical data flow for PostCollection is shown in figure 8.

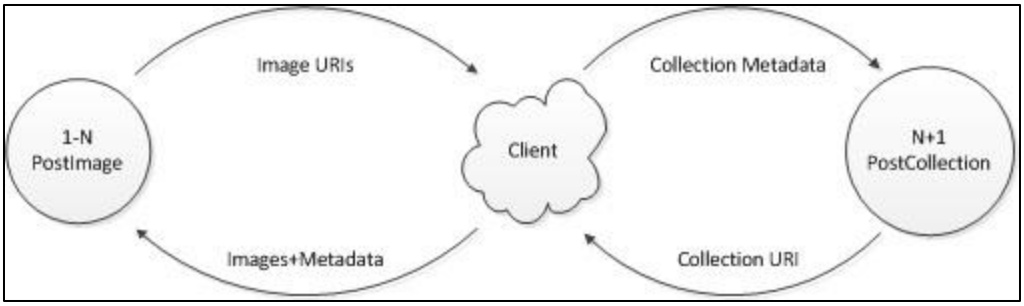


Figure 8. PostCollection.

Request Syntax:
POST http://{host}:{port}/ARLImageProcessing/collections

Input Parameters	Definition	Use
Title	the title of the collection	optional
ImageURI	the URIs of the images/collections in the collection	required

Output Parameters	Definition	Cardinality
string	the server-generated URI of the collection	1..1

```
POST http://{host}:{port}/ARLImageProcessing/collections HTTP/1.1

<?xml version="1.0" encoding="utf-8"?>
<Collection
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns="http://arl.army.milhttp://{host}:{port}/ARLImageProcessing/Collection.xsd">
   <Title>My Collection</Title>
   <ImageURI>http://mule01.lsn.arl:8080http://{host}:{port}/ARLImageProcessing/images/44b8cc45-d991-4c6f-bc23-3086d47e8429</ImageURI>
   <ImageURI>http://mule01.lsn.arl:8080http://{host}:{port}/ARLImageProcessing/images/699ad685-3a72-43d1-b365-d29e9b6b027c</ImageURI>
</Collection>
```

Listing 17. Example PostCollection request.

```
HTTP/1.1 201 Created

<?xml version="1.0" encoding="utf-8"?>
<string>http://mule01.lsn.arl:8080http://{host}:{port}/ARLImageProcessing/collections/eceea312-2138-4038-933e-359f477f2460</string>
```

Listing 18. Example PostCollection response.

4.2 GetCollection

Retrieve a collection's metadata. The data flow for GetCollection is shown in figure 9.

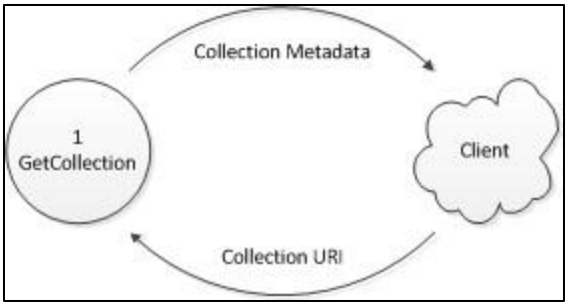


Figure 9. GetCollection.

Request Syntax:
GET http://{host}:{port}/ARLImageProcessing/collections/{collectionID}

Input Parameters	Definition	Use
collectionID	the guid of collection to retrieve	required

Output Parameters	Definition	Cardinality
URI	the URI of the collection	1..1
Title	the title of the collection	0..1
DateCreated	sever-generated timestamp of when the collection was uploaded	1..1
ImageURI	the URIs of the images and/or other collections in the collection	1..N

```
GET http://{host}:{port}/ARLImageProcessing/collections/088d6c9b-8d76-42f6-a0d9-b18825f14ae5 HTTP/1.1
```

Listing 19. Example GetCollection request.

```
HTTP/1.1 200 OK

<?xml version="1.0" encoding="utf-8" ?>
<Collection URI="http://mule01.lsn.arl:8080http://{host}:{port}/ARLImageProcessing/collections/088d6c9b-8d76-42f6-a0d9-b18825f14ae5">
   <Title>Camera SuperResolution</Title>
   <DateCreated>2012-08-29T13:41:34.4395463-04:00</DateCreated>
   <ImageURI>http://mule01.lsn.arl:8080http://{host}:{port}/ARLImageProcessing/images/57a01266-9dee-4344-afb0-2930d6fc3e7c</ImageURI>
   <ImageURI>http://mule01.lsn.arl:8080http://{host}:{port}/ARLImageProcessing/images/18efe828-1426-49f4-82fd-ccd1ef21bc3c</ImageURI>
   <ImageURI>http://mule01.lsn.arl:8080http://{host}:{port}/ARLImageProcessing/images/a7e2de55-c30e-481a-9ae0-887575195173</ImageURI>
</Collection>
```

Listing 20. Example GetCollection response.

4.3 GetCollectionList

Retrieve the metadata for all of the collections. A typical data flow for GetCollectionList is shown in figure 10.

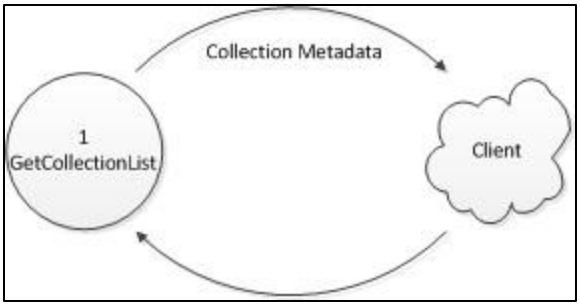


Figure 10. GetCollectionList.

Request Syntax:
GET http://{host}:{port}/ARLImageProcessing/collections

Output Parameters	Definition	Cardinality
Collection	the metadata for a single collection Collection	0..N
Collection.URI	the URI of the collection	1..1
Collection.Title	the title of the collection	0..1
Collection.DateCreated	sever-generated timestamp of when the collection was uploaded	1..1
Collection.ImageURI	the URIs of the images and/or other collections in the collection	1..N

```
GET http://{host}:{port}/ARLImageProcessing/collections HTTP/1.1
```

Listing 21. Example GetCollection request.

```
HTTP/1.1 200 OK

<?xml version="1.0" encoding="utf-8" ?>
<ArrayOfCollection>
  <Collection URI="http://mule01.lsn.arl:8080/ARLImageProcessing/collections/088d6c9b-8d76-42f6-a0d9-b18825f14ae5">
    <Title>Camera SuperResolution</Title>
    <DateCreated>2012-08-29T13:41:34.4395463-04:00</DateCreated>
    <ImageURI>http://mule01.lsn.arl:8080/ARLImageProcessing/images/57a01266-9dee-4344-afb0-2930d6fc3e7c</ImageURI>
    <ImageURI>http://mule01.lsn.arl:8080/ARLImageProcessing/images/18efe828-1426-49f4-82fd-ccd1ef21bc3c</ImageURI>
    <ImageURI>http://mule01.lsn.arl:8080/ARLImageProcessing/images/ed262f3b-d940-4bbf-b9ea-d304cba39196</ImageURI>
    <ImageURI>http://mule01.lsn.arl:8080/ARLImageProcessing/images/56bfe4b5-0503-4c11-9653-b0f8826e2368</ImageURI>
    <ImageURI>http://mule01.lsn.arl:8080/ARLImageProcessing/images/a7e2de55-c30e-481a-9ae0-887575195173</ImageURI>
  </Collection>
  <Collection URI="http://mule01.lsn.arl:8080/ARLImageProcessing/collections/14c76164-b763-4aab-ab06-3312a7e93111">
    <Title>Face</Title>
    <DateCreated>2012-05-24T15:35:51.2496068-04:00</DateCreated>
    <ImageURI>http://mule01.lsn.arl:8080/ARLImageProcessing/images/c0e3a371-4951-4009-b338-506dd3b7ce0a</ImageURI>
    <ImageURI>http://mule01.lsn.arl:8080/ARLImageProcessing/images/f385bf6c-51eb-4ec3-ae05-ec03a8cd55f6</ImageURI>
    <ImageURI>http://mule01.lsn.arl:8080/ARLImageProcessing/images/37a19341-c25c-4f10-9757-b72f4a2a38cc</ImageURI>
    <ImageURI>http://mule01.lsn.arl:8080/ARLImageProcessing/images/4085bc6d-0a7c-4a32-b105-215e1f5b9cca</ImageURI>
  </Collection>
</ArrayOfCollection>
```

Listing 22. Example GetCollection response.

4.4 DeleteCollection

Delete a collection. Note that the associated images themselves are not deleted and any references to the collection in other collections will be left dangling. The data flow for DeleteCollection is show in figure 11.

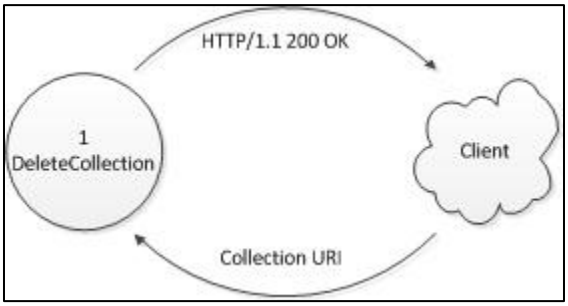


Figure 11. DeleteCollection.

Request Syntax:
DELETE http://{host}:{port}/ARLImageProcessing/collections/{collectionID}

Input Parameters	Definition	Use
collectionID	the guid of the collection	required

```
DELETE http://{host}:{port}/ARLImageProcessing/collections/c734551e-bc64-453b-a361-4dbe17f1f356
HTTP/1.1
```

Listing 23. Example DeleteCollection request.

```
HTTP/1.1 200 OK
```

Listing 24. Example DeleteCollection response.

4.5 PutCollection

Replace a collection's metadata. This is typically used to change the collection's title or add/remove images to/from the collection. Note that a collection's metadata is entirely replaced and not merged or otherwise checked in any manner. To add images to the collection, the client must supply all of the current image URIs as well as any new ones in the metadata. To remove images from the collection, the client must supply all the current image URIs except the ones to remove. Use this with caution since even the collection's URI can be changed or deleted with this method, resulting in a corrupt collection. The data flow for PutCollection is shown in figure 12.

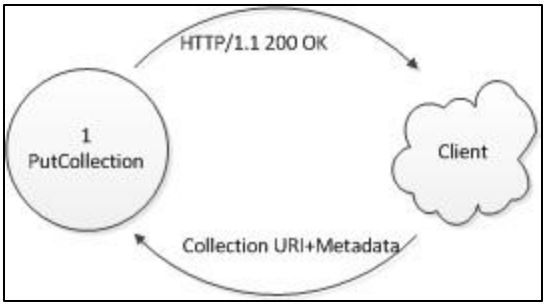


Figure 12. PutCollection.

Request Syntax:
PUT /http://{host}:{port}/ARLImageProcessing/collections/{collectionID}

Input Parameters	Definition	Use
collectionID	the guid of collection's metadata to replace	required
DateCreated	the date and time the collection was created	copy from current metadata to preserve the orignal creation date or replace with current date/time to preserve the modification timestamp
URI	the URI of the collection	copy from current meatadata
Title	the title of the collection	optional
ImageURI	the URIs of the images in the collection	complete list of image URIs in the collection

```
PUT /http://{host}:{port}/ARLImageProcessing/collections/c734551e-bc64-453b-a361-4dbe17f1f356 HTTP/1.1

<?xml version="1.0" encoding="utf-8"?>
<Collection URI="http://mule01.lsn.arl:8080/ARLImageProcessing/collections/c734551e-bc64-453b-a361-
4dbe17f1f356">
   <Title>My Collection</Title>
   <DateCreated>2012-08-29T14:18:45.9298657-04:00</DateCreated>
   <ImageURI>http://mule01.lsn.arl:8080/ARLImageProcessing/images/44b8cc45-d991-4c6f-bc23-
3086d47e8429</ImageURI>
   <ImageURI>http://mule01.lsn.arl:8080/ARLImageProcessing/images/699ad685-3a72-43d1-b365-
d29e9b6b027c</ImageURI>
</Collection>
```

Listing 25. Example PutCollection request.

```
HTTP/1.1 200 OK
```

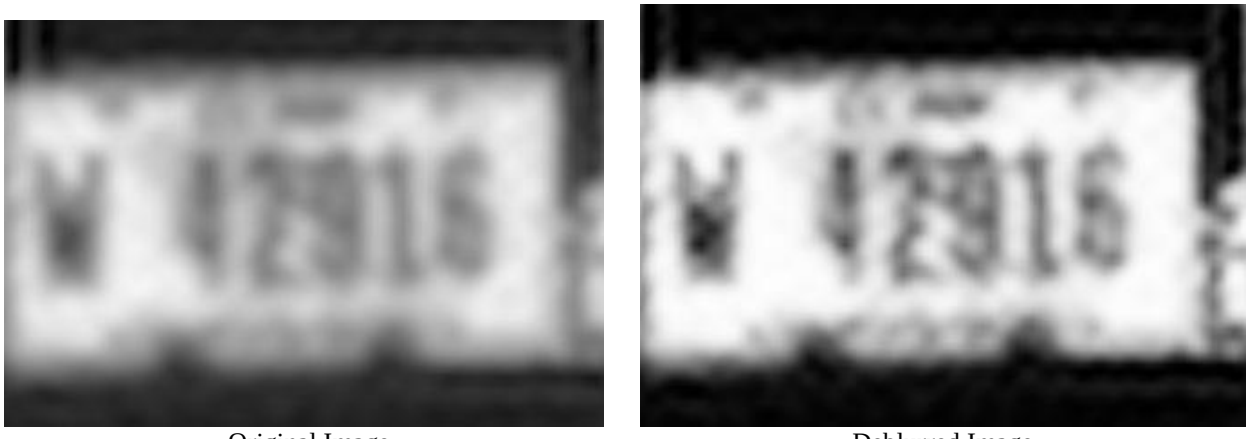
Listing 26. Example PutCollection response.

5. Image Deblurring Services

In this section, we describe the Web services available for deblurring an image. The algorithm used here was developed in the ARL's SEDD Image Processing Branch (Young, Driggers and Teaney, et al. 2007). The following abstract sums up the algorithm and its use:

> "This report proposes a practical sensor deblur filtering method for images that are contaminated with noise. A sensor blurring function is usually modeled via a Gaussian-like function having a bell shape. The straightforward inverse function results in magnification of noise at the high frequencies. In order to address this issue, we apply a special spectral window to the inverse blurring function. This special window is called the *power* window, which is a Fourier-based smoothing window that preserves most of the spatial frequency components in the passband and attenuates quickly at the transition-band. The power window is differentiable at the transition point which gives a desired smooth property and limits the ripple effect. Utilizing properties of the power window, we design the deblurring filter adaptively by estimating energy of the signal and noise of the image to determine the passband and transition-band of the filter. The deblurring filter design criteria are: a) filter magnitude is less than one at the frequencies where the noise is stronger than the desired signal (transition-band); and b) filter magnitude is greater than one at the other frequencies (passband). Therefore, the adaptively designed deblurring filter is able to deblur the image by a desired amount based on the estimated or known blurring function while suppressing the noise in the output image." (Young, Driggers and Teaney, et al. 2007)

An example of deblurring is shown in image 2.



Original Image Deblurred Image

Image 2. Deblurring.

5.1 PostImageDeblurred

Perform a deblurring of the image specified ParentURI and return the server-generated URI of the deblurred image. The typical steps involved with deblurring an image are to first post a base image and then post the deblurring parameters. If the base image already exists, then the first step can be skipped. The data flow for PostImageDeblurred is shown in figure 13.

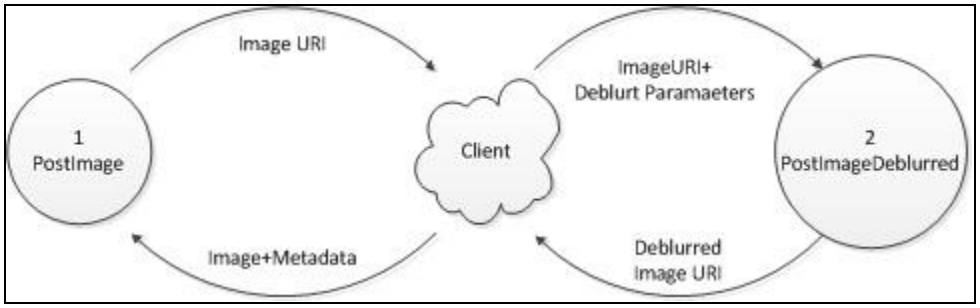


Figure 13. PostImageDeblurred.

The deblur filter assumes the blurring is Gaussian. Two parameters are used to adjust the deblur filter: MG and Alpha. MG is the magnitude of the inverse Gaussian and can be varied between 1–100. Alpha is the ratio of the standard deviation of the Gaussian filter to the maximum polar frequency—it controls the bandwidth of the Gaussian filter. Alpha can be varied between 0 and 1. For an initial setting, use alpha=0.25 and mg=5. Decreasing alpha sharpens the image further; however, artifacts may appear with low values of Alpha.

Request Syntax:
POST http://{host}:{port}/ARLImageProcessing/images/deblurred

Input Parameters	Definition	Use
Title	the title of the image	optional
ParentURI	the URI of the image to deblur	required
DeblurParameters.Alpha	the ratio of the standard deviation of the Gaussian filter to the maximum polar frequency	required
DeblurParameters.MG	the magnitude of the inverse Gaussian	required

Output Parameters	Definition	Cardinality
string	the server-generated URI of the deblurred image	1..1

```
POST http://{host}:{port}/ARLImageProcessing/images/deblurred HTTP/1.1

<?xml version="1.0" encoding="utf-8"?>
<Metadata xmlns:inkml="http://www.w3.org/2003/InkML"
xmlns="http://arl.army.mil/http://{host}:{port}/ARLImageProcessing/Metadata.xsd">
   <Title>Contrast-Enhanced Face</Title>
   <ParentURI>http://mule01.lsn.arl:8080/ARLImageProcessing/images/45695db1-9605-438e-a7e5-
d8eb2d4c75b1</ParentURI>
   <DeblurParameters Alpha="0.11" MG="9"/>
</Metadata>
```

Listing 27. Example PostImageDeblurred request.

```
HTTP/1.1 201 Created

<?xml version="1.0" encoding="utf-8"?>
<string>http://mule01.lsn.arl:8080http://{host}:{port}/ARLImageProcessing/images/deblurred/d35c6f84-9cfc-4b82-
8b1b-cf11d8a02641</string>
```

Listing 28. Example PostImageDeblurred response.

5.2 GetImageDeblurred

Retrieve a deblurred image and its metadata. The data flow for GetImageDeblurred is shown in figure 14.

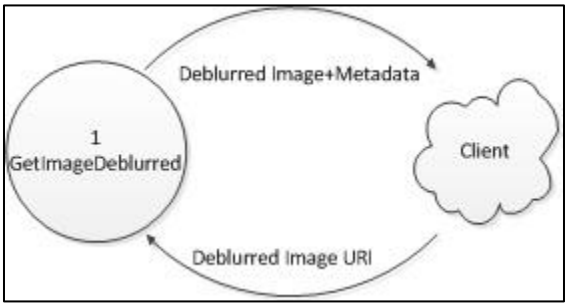


Figure 14. GetImageDeblurred.

Request Syntax:
GET http://{host}:{port}/ARLImageProcessing/images/deblurred/{imageID}

Input Parameters	Definition	Use
imageID	the guid of the deblurred image to retrieve	required

Output Parameters	Definition	Cardinality
URI	the URI of the deblurred image	1..1
MediaType	"image/png"	0..1
Title	the title of the image	0..1
DateCreated	the date and time the image was created	1..1
ParentURI	the URI of the parent image	1..1
DeblurParameters	the deblur parameters for this image	1..1
DeblurParameters.Alpha	the ratio of the standard deviation of the Gaussian filter to the maximum polar	1..1

	frequency	
DeblurParameters.MG	the magnitude of the inverse Gaussian	1..1
Annotations	textual and graphical annotations	0..1
Annotations.Text	any textual annotations for the image	0..N
Annotations.Text.X	X pixel position for the start of the text	1..1
Annotations.Text.Y	Y pixel position for the start of the text	1..1
Annotations.Text.Font	optional font the text	0..1
Annotations.Text.Size	optional size for the font	0..1
Annotations.Text.Color	optional ARGB color for the text	0..1
Annotations.ink	any InkML markup for the image	0..1
Media	the base 64 ASCII encoding of the deblurred image	1..1

```
GET /http://{host}:{port}/ARLImageProcessing/images/deblurred/566be980-6f55-424b-b790-abc653210db9
HTTP/1.1
```

Listing 29. Example GetImageDeblurred request.

```
HTTP/1.1 200 OK

<?xml version="1.0" encoding="utf-8" ?>
<Metadata URI="http://mule01.lsn.arl:8080/ARLImageProcessing/images/deblurred/566be980-6f55-424b-b790-abc653210db9">
   <MediaType>image/png</MediaType>
   <Title>Deblurred Super Resolved Face</Title>
   <DateCreated>2012-08-29T13:16:42.3125255-04:00</DateCreated>
   <ParentURI>http://mule01.lsn.arl:8080/ARLImageProcessing/collections/superresolved/16dd1114-b0b8-43b6-84c8-236e304bdf23</ParentURI>
   <DeblurParameters Alpha="0.2" MG="15" />
   <Media>iVBORw0KGgoAAAANSUhEUgAAAVAAA ...</Media>
</Metadata>
```

Listing 30. Example GetImageDeblurred response.

5.3 GetImageDeblurredMetadata

Retrieve a deblurred image's metadata. The data flow for GetImageDeblurredMetadata is shown in figure 15.

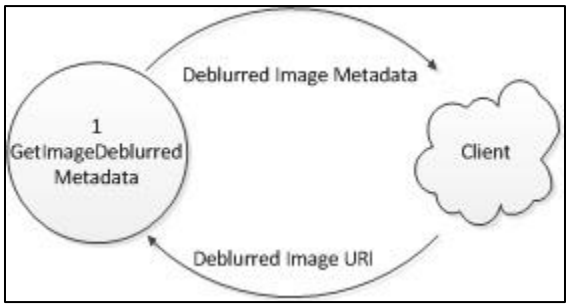


Figure 15. GetImageDeblurredMetadata.

Request Syntax:
GET /http://{host}:{port}/ARLImageProcessing/images/deblurred/{imageID}/metadata

Input Parameters	Definition	Use
imageID	the guid of the deblurred image metadata to retrieve	required

Output Parameters	Definition	Cardinality
URI	the URI of the deblurred image	1..1
MediaType	"image/png"	0..1
Title	the title of the image	0..1
DateCreated	the date and time the image was created	1..1
ParentURI	the URI of the parent image	1..1
DeblurParameters	the deblur parameters for this image	1..1
DeblurParameters.Alpha	the ratio of the standard deviation of the Gaussian filter to the maximum polar frequency	1..1
DeblurParameters.MG	the magnitude of the inverse Gaussian	1..1
Annotations	textual and graphical annotations	0..1
Annotations.Text	any textual annotations for the image	0..N
Annotations.Text.X	X pixel position for the start of the text	1..1
Annotations.Text.Y	Y pixel position for the start of the text	1..1
Annotations.Text.Font	optional font the text	0..1
Annotations.Text.Size	optional size for the font	0..1
Annotations.Text.Color	optional ARGB color for the text	0..1
Annotations.ink	any InkML markup for the image	0..1

```
GET http://{host}:{port}/ARLImageProcessing/images/deblurred/0476ef6a-bbb4-41bd-bbd5-
f8117a470514/metadata HTTP/1.1
```

Listing 31. Example GetImageDeblurredMetadata request.

```
HTTP/1.1 200 OK

<?xml version="1.0" encoding="utf-8" ?>
<Metadata URI="http://mule01.lsn.arl:8080/ARLImageProcessing/images/deblurred/0b80a8af-8148-4c51-8042-
0fb25091835e"
   xmlns="http://arl.army.mil/http://{host}:{port}/ARLImageProcessing/Metadata.xsd"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <MediaType>image/png</MediaType>
   <DateCreated>2012-08-28T16:03:43.9699741-04:00</DateCreated>
   <ParentURI>http://mule01.lsn.arl:8080/ARLImageProcessing/collections/superresolved/bc205091-232c-4559-
9f20-cc79c0941bb1</ParentURI>
   <DeblurParameters Alpha="0.11" MG="9"/>
</Metadata>
```

Listing 32. Example GetImageDeblurredMetadata response.

5.4 GetImageDeblurredOverlay

Retrieve a transparent image consisting of just the textual and graphical annotations of a deblurred image. Applications that do not want to render the annotations and markup themselves

can call this service to retrieve a transparent image of just the markup for a deblurred image and then overlay it on top of the image to see the annotations. The data flow for GetImageDeblurredOverlay is shown in figure 16.

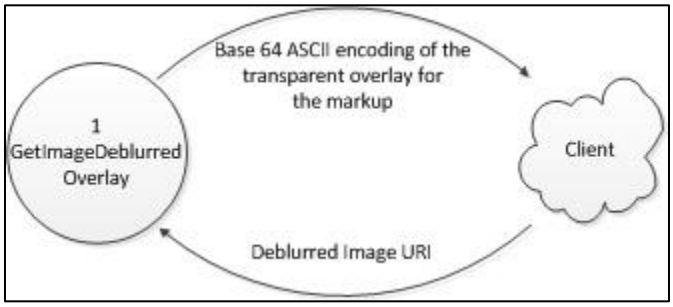


Figure 16. GetImageDeblurredOverlay.

Request Syntax:
GET /http://{host}:{port}/ARLImageProcessing/images/deblurred/{imageID}/overlay

Input Parameters	Definition	Use
imageID	the guid of the annotated deblurred image	required

Output Parameters	Definition	Cardinality
base64Binary	the base 64 ASCII encoding of the overlay image	1..1

```
GET http://{host}:{port}/ARLImageProcessing/images/deblurred/9e635628-b064-4240-8dc0-
c4ec1750f874/overlay HTTP/1.1
```

Listing 33. Example GetImageDeblurredOverlay request.

```
HTTP/1.1 200 OK

<?xml version="1.0" encoding="utf-8" ?>
<base64Binary>iVBORw0KGgoAAAANSUhE ...</base64Binary>
```

Listing 34. Example GetImageDeblurredOverlay response.

5.5 GetImageDeblurredList

Retrieve the metadata associated with all of the deblurred images. The data flow for GetImageDeblurredList is shown in figure 17.

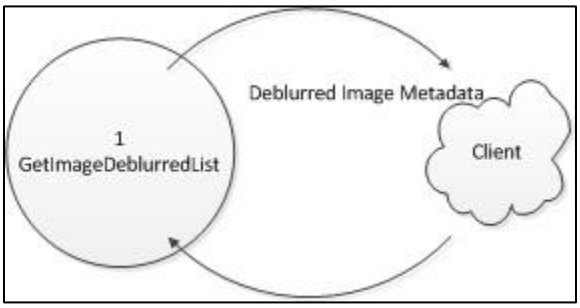


Figure 17. GetImageDeblurredList.

Request Syntax:

GET http://{host}:{port}/ARLImageProcessing/images/deblurred

Output Parameters	Definition	Cardinality
Metadata	the metadata for a single deblurred image	0..N
Metadata.URI	the URI of the deblurred image	1..1
Metadata.MediaType	"image/png"	0..1
Metadata.Title	the title of the image	0..1
Metadata.DateCreated	the date and time the image was created	1..1
Metadata.ParentURI	the URI of the parent image	1..1
Metadata.DeblurParameters	the deblur parameters for this image	1..1
Metadata.DeblurParameters.Alpha	the ratio of the standard deviation of the Gaussian filter to the maximum polar frequency	1..1
Metadata.DeblurParameters.MG	the magnitude of the inverse Gaussian	1..1
Metdata.Annotations	textual and graphical annotations	0..1
Metadata Annotations.Text	any textual annotations for the image	0..N
Metadata.Annotations.Text.X	X pixel position for the start of the text	1..1
Metadata.Annotations.Text.Y	Y pixel position for the start of the text	1..1
Metadata.Annotations.Text.Font	optional font the text	0..1
Metadata.Annotations.Text.Size	optional size for the font	0..1
Metadata.Annotations.Text.Color	optional ARGB color for the text	0..1
Metadata.Annotations.ink	any InkML markup for the image	0..1

```
GET http://{host}:{port}/ARLImageProcessing/images/deblurred HTTP/1.1
```

Listing 35. Example GetImageDeblurredList request.

```
HTTP/1.1 200 OK

<?xml version="1.0" encoding="utf-8" ?>
<ArrayOfMetadata>
  <Metadata URI="http://mule01.lsn.arl:8080/ARLImageProcessing/images/deblurred/0476ef6a-bbb4-41bd-bbd5-f8117a470514">
    <MediaType>image/png</MediaType>
    <Title>Deblurred Face</Title>
    <DateCreated>2012-08-29T14:15:57.2311694-04:00</DateCreated>
    <ParentURI>http://mule01.lsn.arl:8080/ARLImageProcessing/images/9e635628-b064-4240-8dc0-c4ec1750f874</ParentURI>
    <DeblurParameters Alpha="0.11" MG="9"/>
  </Metadata>
  <Metadata URI="http://mule01.lsn.arl:8080/ARLImageProcessing/images/deblurred/0b3f4a63-2c28-4000-b3e6-1bde87ef5055">
    <MediaType>image/png</MediaType>
    <DateCreated>2012-09-10T16:27:03.9392295-04:00</DateCreated>
    <ParentURI>http://mule01.lsn.arl:8080/ARLImageProcessing/images/contrastenhanced6decf012-35a8-44f9-aa70-8ff1cc98f5d5</ParentURI>
    <DeblurParameters Alpha="0.16" MG="5"/>
  </Metadata>
</ArrayOfMetadata>
```

Listing 36. Example GetImageDeblurredList response.

5.6 DeleteImageDeblurred

Delete a deblurred image. Note that any references to the image in any collections will be left dangling. The data flow for DeleteImageDeblurred is shown in figure 18.

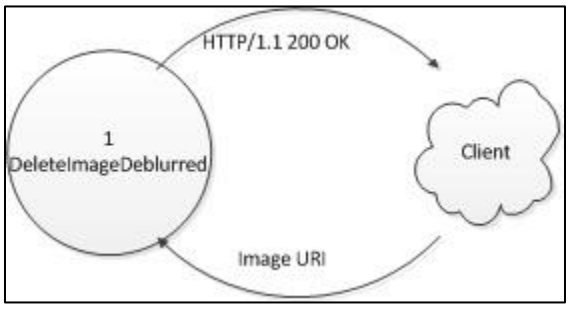


Figure 18. DeleteImageDeblurred.

Request Syntax:
DELETE http://{host}:{port}/ARLImageProcessing/images/deblurred/{imageID}

Input Parameters	Definition	Use
imageID	the guid of the deblurred image to delete	required

```
DELETE http://{host}:{port}/ARLImageProcessing/images/deblurred/c734551e-bc64-453b-a361-4dbe17f1f356 HTTP/1.1
```

Listing 37. Example DeleteImageDeblurred request.

```
HTTP/1.1 200 OK
```

Listing 38. Example DeleteImageDeblurred response.

5.7 PutImageDeblurredMetadata

Replace a deblurred image's metadata. Note that the image's metadata is entirely replaced and not merged or otherwise checked in any manner. Use cautiously since even the image's URI can be changed or deleted with this method, resulting in not only a corrupt image, but also corrupting any collections it belongs to. To add to the metadata, the current metadata should first be retrieved and copied into the new metadata. This method is typically used to add textual annotations and ink markup to an existing image. The data flow for PutImageDeblurredMetadata is shown in figure 19.

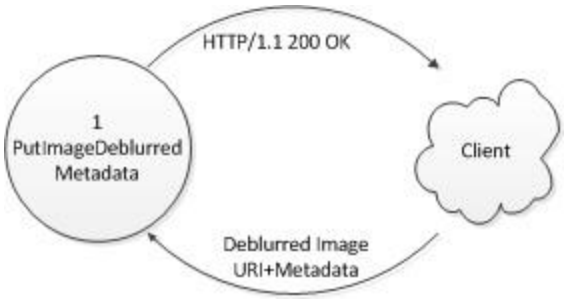


Figure 19. PutImageDeblurredMetadata.

Request Syntax:

PUT /http://{host}:{port}/ARLImageProcessing/images/deblurred/{imageID}

Input Parameters	Definition	Use
imageID	the guid of deblurred image's metadata to replace	required
URI	the URI of the deblurred image	copy from current metatadata
MediaType	the mime type of the deblurred image	copy from current metatadata
Title	the title of the image	optional
DateCreated	the date and time the image was deblurred	copy from current metadata to preserve the orignal creation date or replace with current date/time to preserve the modification timestamp
ParentURI	the URI of parent image which was deblurred	copy from current metatadata
DeblurParameters.Alpha	the ratio of the standard deviation of the Gaussian filter to the maximum polar frequency	copy from current metatadata
DeblurParameters.MG	the magnitude of the inverse Gaussian	copy from current metatadata
Annotations	textual and graphical annotations	0..1
Annotations.Text	any textual annotations for the image	optional
Annotations.Text.X	X pixel position for the start of the text	required
Annotations.Text.Y	Y pixel position for the start of the text	required
Annotations.Text.Font	optional font the text	optional
Annotations.Text.Size	optional size for the font	optional
Annotations.Text.Color	optional ARGB color for the text	optional
Annotations.ink	any InkML markup for the image	optional

```
PUT /http://{host}:{port}/ARLImageProcessing/images/deblurred/07659a20-6ac3-42bf-a6a4-93dc26527a39
HTTP/1.1

<?xml version="1.0" encoding="utf-8" ?>
<Metadata URI="http://mule01.lsn.arl:8080/ARLImageProcessing/images/deblurred/07659a20-6ac3-42bf-a6a4-
93dc26527a39">
   <MediaType>image/png</MediaType>
   <Title>Deblurred Face</Title>
   <DateCreated>2012-08-29T14:18:45.9298657-04:00</DateCreated>
   <ParentURI>http://mule01.lsn.arl:8080/ARLImageProcessing/images/9618d68b-0f46-4f28-8ef8-
7dd4765d64ad</ParentURI>
   <DeblurParameters Alpha="0.11" MG="9"/>
   <Annotations>
      <Text X="191" Y="186" Font="Segoe360" Size="12" Color="#FFFF0000">Some Text</Text>
      <Text X="227" Y="315" Font="Segoe360" Size="12" Color="#FFFF0000">Some More Text</Text>
      <ink>
         <definitions>
            <brush xml:id="1">
               <brushProperty name="width" value="2.00314960629921" />
               <brushProperty name="height" value="2.00314960629921" />
               <brushProperty name="color" value="#FF00FFFF" />
               <brushProperty name="tip" value="ellipse" />
            </brush>
         </definitions>
         <trace brushRef="1">38.5566032170582 192.222980271295, 39.4092523017537 191.65161538904,
39.4069967680897 190.702712261337</trace>
      </ink>
   </Annotations>
</Metadata>
```

Listing 39. Example PutImageDeblurredMetadata request.

```
HTTP/1.1 200 OK
```

Listing 40. Example PutImageDeblurredMetadata response.

6. Image Contrast Enhancement Services

In this section, we describe the Web services available for enhancing the contrast of an image. The algorithm used here was developed in the ARL's SEDD Image Processing Branch (Maschel and Young 2012). The abstract following abstract sums up the algorithm and its use:

"In surveillance applications, the visibility of details within an image is necessary to ensure detection. However, bright spots in images can occupy most of the dynamic range of the sensor, causing lower energy details to appear dark and difficult to see. In addition, shadows from structures such as buildings or bridges obscure features within the image, further limiting contrast. Dynamic range compression and contrast enhancement algorithms can be used to improve the visibility of these low energy details. In this paper, we propose a locally adaptive contrast enhancement algorithm based on the multi-scale wavelet transform to compress the dynamic range of images as well as increase the visibility of details obscured by shadows. Using an edge detector as the mother wavelet, this algorithm operates by increasing the gain of low energy gradient magnitudes provided by the wavelet transform, while simultaneously decreasing the gain of higher energy gradient magnitudes. Limits on the amount of gain imposed are set locally to prevent the over-enhancement of noise. The results of using the proposed method on aerial images show that this method outperforms common methods in its ability to enhance small details while simultaneously preventing ringing artifacts and noise over-enhancement." (Maschel and Young 2012)

An example of a contrast-enhanced image is shown in image 3.

Original Image | Contrast-enhanced Image

Image 3. Contrast enhancement.

6.1 PostImageContrastEnhanced

Perform a contrast enhancement of the image specified ParentURI and return the server-generated URI of the enhanced image. The typical steps involved with enhancing the contrast of an image are to first post a base image and then post the enhancement parameters. If the base image already exists, then the first step can be skipped. The data flow for PostImageContrastEnhanced is shown in figure 20.

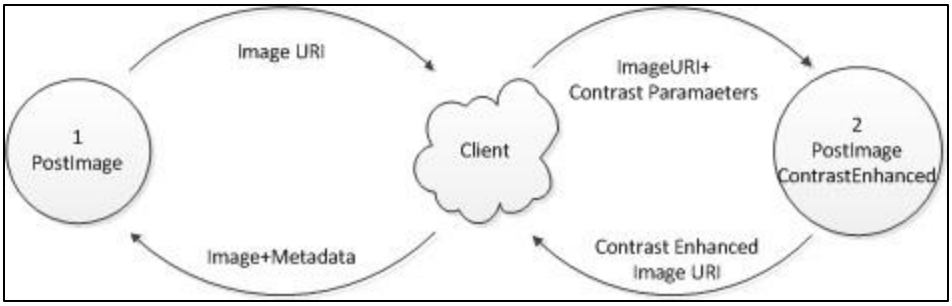


Figure 20. PostImageContrastEnhanced.

The PercentageLow and PercentageHigh contrast enhancement parameters determine the lower tail and upper tail of the accumulated histogram, respectively. If lower pixel values in the image are saturated, start by setting PercentageLow=0.1 and PercentageHigh=0.001. If higher pixel values in the image are saturated, start by setting PercentageLow=0.001 and PercentageHigh=0.1.The possible range of both of these parameters is between 0 and 1, under the constraint that the sum of two must always be less than 1 (i.e., the lower tail cannot overlap with the upper tail).

Request Syntax:
POST http://{host}:{port}/ARLImageProcessing/collections/contrastenhanced

Input Parameters	Definition	Use
Title	the title of the image	optional
ParentURI	the URI of the image to contrast enhance	required
ContrastEnhancementParameters .PercentageLow	lower tail of the accumulated histogram	required
ContrastEnhancementParameters .PercentageHigh	upper tail of the accumulated histogram	required

Output Parameters	Definition	Cardinality
string	the server-generated URI of the contrast enhanced image	1..1

```
POST /http://{host}:{port}/ARLImageProcessing/collections/contrastenhanced HTTP/1.1

<?xml version="1.0" encoding="utf-8"?>
<Metadata xmlns:inkml="http://www.w3.org/2003/InkML"
xmlns="http://arl.army.mil/http://{host}:{port}/ARLImageProcessing/Metadata.xsd">
   <Title>Contrast-Enhanced Face</Title>
   <ParentURI>http://mule01.lsn.arl:8080/ARLImageProcessing/images/45695db1-9605-438e-a7e5-
d8eb2d4c75b1</ParentURI>
   <ContrastEnhancementParameters PercentageLow="0.5" PercentageHigh="0.002" />
</Metadata>
```

Listing 41. Example PostImageContrastEnhanced request.

```
HTTP/1.1 201 Created

<?xml version="1.0" encoding="utf-8"?>
<string>http://mule01.lsn.arl:8080http://{host}:{port}/ARLImageProcessing/images/contrastenhanced/d35c6f84-9cfc-4b82-8b1b-cf11d8a02641</string>
```

Listing 42. Example PostImageContrastEnhanced response.

6.2 GetImageContrastEnhanced

Retrieve a contrast-enhanced image and its metadata. The data flow for GetImageContrastEnhanced is shown in figure 21.

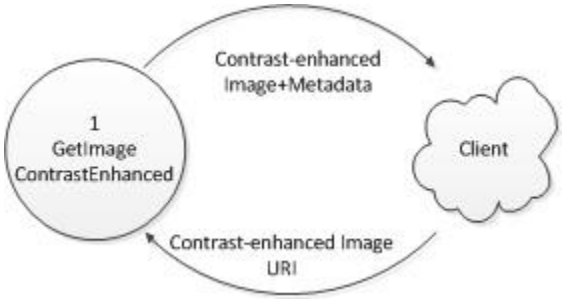


Figure 21. GetImageContrastEnhanced.

Request Syntax:
GET http://{host}:{port}/ARLImageProcessing/images/contrastenhanced/{imageID}

Input Parameters	Definition	Use
imageID	the guid of the contrast-enhanced image to retrieve	required

Output Parameters	Definition	Cardinality
URI	the URI of the contrast-enhanced image	1..1
MediaType	"image/png"	0..1
Title	the title of the image	0..1
DateCreated	the date and time the image was created	1..1
ParentURI	the URI of the parent image	1..1
ContrastEnhancementParameters	the parameters used for contrast enhancement	1..1
ContrastEnhancementParameters .PercentageLow	lower tail of the accumulated histogram	1..1
ContrastEnhancementParameters .PercentageHigh	upper tail of the accumulated histogram	1..1
Annotations	textual and graphical annotations	0..1
Annotations.Text	any textual annotations for the image	0..N
Annotations.Text.X	X pixel position for the start of the text (required)	1...1
Annotations.Text.Y	Y pixel position for the start of the text (required)	1..1
Annotations.Text.Font	optional font the text	0..1
Annotations.Text.Size	optional size for the font	0..1
Annotations.Text.Color	optional ARGB color for the text	0..1

Annotations.ink	any InkML markup for the image	0..1
Media	the base 64 ASCII encoding of the contrast-enhanced image	1..1

```
GET http://{host}:{port}/ARLImageProcessing/images/contrastenhanced/0476ef6a-bbb4-41bd-bbd5-f8117a470514
HTTP/1.1
```

Listing 43. Example GetImageContrastEnhanced request.

```
HTTP/1.1 200 OK

<?xml version="1.0" encoding="utf-8" ?>
<Metadata URI="http://mule01.lsn.arl:8080/ARLImageProcessing/images/contrastenhanced/0476ef6a-bbb4-41bd-
bbd5-f8117a470514">
   <MediaType>image/png</MediaType>
   <Title>Contrast Enhanced Face</Title>
   <DateCreated>2012-08-29T14:15:57.2311694-04:00</DateCreated>
   <ParentURI>http://mule01.lsn.arl:8080/ARLImageProcessing/images/9e635628-b064-4240-8dc0-
c4ec1750f874</ParentURI>
   <ContrastEnhancementParameters PercentageLow="0.5" PercentageHigh="0.002" />
   <Media>iVBORw0KGgoAAAANSUhE ...</Media>
</Metadata>
```

Listing 44. Example GetImageContrastEnhanced response.

6.3 GetImageContrastEnhancedMetadata

Retrieve a contrast-enhanced image's metadata. The data flow for GetImageContrastEnhancedMetadata is shown in figure 22.

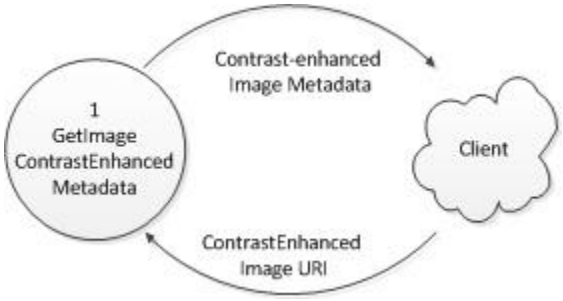


Figure 22. GetImageContrastEnhancedMetadata.

Request Syntax:
GET /http://{host}:{port}/ARLImageProcessing/images/contrastenhanced/{imageID}/metadata

Input Parameters	Definition	Use
imageID	the guid of the contrast-enhanced image metadata to retrieve	required

Output Parameters	Definition	Cardinality
URI	the URI of the contrast-enhanced image	1..1
MediaType	"image/png"	0..1
Title	the title of the image	0..1
DateCreated	the date and time the image was created	1..1
ParentURI	the URI of the parent image	1..1
ContrastEnhancementParameters	the parameters used for contrast enhancement	1..1
ContrastEnhancementParameters.PercentageLow	lower tail of the accumulated histogram	1..1
ContrastEnhancementParameters.PercentageHigh	upper tail of the accumulated histogram	1..1
Annotations	textual and graphical annotations	0..1
Annotations.Text	any textual annotations for the image	0..N
Annotations.Text.X	X pixel position for the start of the text (required)	1...1
Annotations.Text.Y	Y pixel position for the start of the text (required)	1..1
Annotations.Text.Font	optional font the text	0..1
Annotations.Text.Size	optional size for the font	0..1
Annotations.Text.Color	optional ARGB color for the text	0..1
Annotations.ink	any InkML markup for the image	0..1

```
GET http://{host}:{port}/ARLImageProcessing/images/contrastenhanced/0476ef6a-bbb4-41bd-bbd5-f8117a470514/metadata HTTP/1.1
```

Listing 45. Example GetImageContrastEnhancedMetadata request.

```
HTTP/1.1 200 OK

<?xml version="1.0" encoding="utf-8" ?>
<Metadata URI="http://mule01.lsn.arl:8080/ARLImageProcessing/images/contrastenhanced/0476ef6a-bbb4-41bd-bbd5-f8117a470514">
   <MediaType>image/png</MediaType>
   <Title>Contrast Enhanced Face</Title>
   <DateCreated>2012-08-29T14:15:57.2311694-04:00</DateCreated>
   <ParentURI>http://mule01.lsn.arl:8080/ARLImageProcessing/images/9e635628-b064-4240-8dc0-c4ec1750f874</ParentURI>
   <ContrastEnhancementParameters PercentageLow="0.5" PercentageHigh="0.002" />
</Metadata>
```

Listing 46. Example GetImageContrastEnhancedMetadata response.

6.4 GetImageContrastEnhancedOverlay

Retrieve a transparent image consisting of just the textual and graphical annotations of a contrast-enhanced image. Applications that do not want to render the annotations and markup themselves can call this service to retrieve a transparent image of just the markup for a contrast-enhanced image and then overlay it on top of the image to see the annotations. The data flow for GetImageContrastEnhancedOverlay is shown in figure 23.

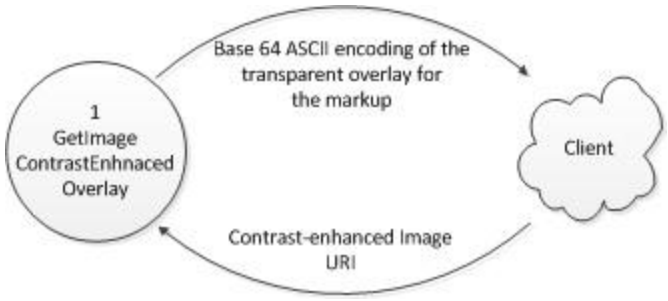


Figure 23. GetImageContrastEnhancedOverlay.

Request Syntax:
GET http://{host}:{port}/ARLImageProcessing/images/contrastenhanced/{imageID}/overlay

Input Parameters	Definition	Use
imageID	the guid of the annotated contrast-enhanced image	required

Output Parameters	Definition	Cardinality
base64Binary	the base 64 ASCII encoding of the overlay image	1..1

```
GET http://{host}:{port}/ARLImageProcessing/images/contrastenhanced/0476ef6a-bbb4-41bd-bbd5-f8117a470514/overlay HTTP/1.1
```

Listing 47. Example GetImageContrastEnhancedOverlay request.

```
HTTP/1.1 200 OK

<?xml version="1.0" encoding="utf-8" ?>
<base64Binary>iVBORw0KGgoAAAANSUhE ...</base64Binary>
```

Listing 48. Example GetImageContrastEnhancedOverlay response.

6.5 GetImageContrastEnhancedList

Retrieve the metadata associated with all of the contrast-enhanced images. The data flow for GetImageContrastEnhancedList is shown in figure 24.

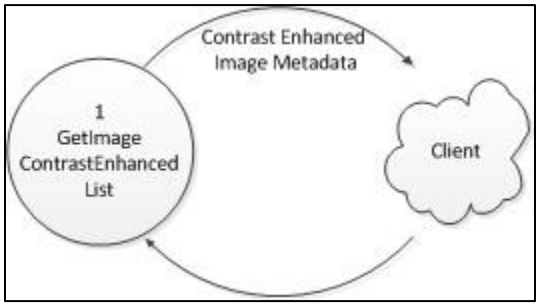


Figure 24. GetImageContrastEnhancedList.

Request Syntax:
GET /http://{host}:{port}/ARLImageProcessing/images/contrastenhanced

Output Parameters	Definition	Cardinality
Metadata	the metadata for a single contrast-enhanced image	0..N
Metadata.URI	the URI of the contrast-enhanced image	1..1
Metadata.MediaType	"image/png"	0..1
Metadata.Title	the title of the image	0..1
Metadata.DateCreated	the date and time the image was created	1..1
Metadata.ParentURI	the URI of the parent image	1..1
Metadata.ContrastEnhancementParameters	the parameters used for contrast enhancement	1..1
Metadata.ContrastEnhancementParameters .PercentageLow	lower tail of the accumulated histogram	1..1
Metadata.ContrastEnhancementParameters .PercentageHigh	upper tail of the accumulated histogram	1..1
Metadata.Annotations	textual and graphical annotations	0..1
Metadata.Annotations.Text	any textual annotations for the image	0..N
Metadata.Annotations.Text.X	X pixel position for the start of the text	1..1
Metadata.Annotations.Text.Y	Y pixel position for the start of the text	1..1
Metadata.Annotations.Text.Font	optional font the text	0..1
Metadata.Annotations.Text.Size	optional size for the font	0..1
Metadata.Annotations.Text.Color	optional ARGB color for the text	0..1
Metadata.Annotations.ink	any InkML markup for the image	0..1

```
GET /http://{host}:{port}/ARLImageProcessing/images/contrastenhanced HTTP/1.1
```

Listing 49. Example GetImageContrastEnhancedList request.

```
HTTP/1.1 200 OK
<?xml version="1.0" encoding="utf-8" ?>
<ArrayOfMetadata>
  <Metadata URI="http://mule01.lsn.arl:8080/ARLImageProcessing/images/contrastenhanced/0476ef6a-bbb4-41bd-bbd5-f8117a470514">
    <MediaType>image/png</MediaType>
    <Title>Contrast Enhanced Face</Title>
    <DateCreated>2012-08-29T14:15:57.2311694-04:00</DateCreated>
    <ParentURI>http://mule01.lsn.arl:8080/ARLImageProcessing/images/9e635628-b064-4240-8dc0-c4ec1750f874</ParentURI>
    <ContrastEnhancementParameters PercentageLow="0.5" PercentageHigh="0.002"/>
  </Metadata>
  <Metadata URI="http://mule01.lsn.arl:8080/ARLImageProcessing/images/contrastenhanced/0b3f4a63-2c28-4000-b3e6-1bde87ef5055">
    <MediaType>image/png</MediaType>
    <DateCreated>2012-09-10T16:27:03.9392295-04:00</DateCreated>
    <ParentURI>http://mule01.lsn.arl:8080/ARLImageProcessing/images/6decf012-35a8-44f9-aa70-8ff1cc98f5d5</ParentURI>
    <ContrastEnhancementParameters PercentageLow="0.5" PercentageHigh="0.002"/>
  </Metadata>
</ArrayOfMetadata>
```

Listing 50. Example GetImageContrastEnhnancedList response.

6.6 DeleteImageContrastEnhanced

Delete a contrast-enhanced image. Note that any references to the image in any collections will be left dangling. The data flow for DeleteImageContrastEnhanced is shown in figure 25.

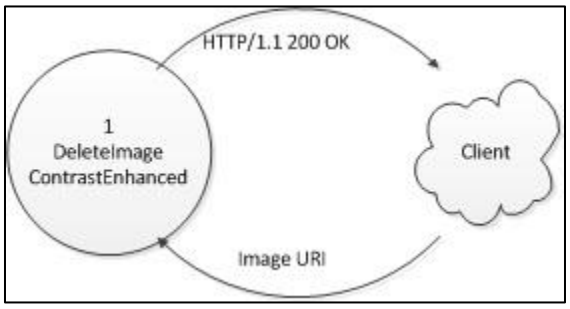


Figure 25. DeleteImageContrastEnhanced.

Request Syntax:
DELETE http://{host}:{port}/ARLImageProcessing/images/deblurred/{imageID}

Input Parameters	Definition	Use
imageID	the guid of the contrast-enhanced image to delete	required

```
DELETE /http://{host}:{port}/ARLImageProcessing/images/contrastenhanced/c734551e-bc64-453b-a361-4dbe17f1f356 HTTP/1.1
```

Listing 51. Example DeleteImageContrastEnhanced request.

```
HTTP/1.1 200 OK
```

Listing 52. Example DeleteImageContrastEnhanced response.

6.7 PutImageContrastEnhancedMetadata

Replace a contrast-enhanced image's metadata. Note that the image's metadata is entirely replaced and not merged or otherwise checked in any manner. Use cautiously since even the image's URI can be changed or deleted with this method, resulting in not only a corrupt image, but also corrupting any collections it belongs to. To add to the metadata, the current metadata should first be retrieved and copied into the new metadata. This method is typically used to add textual annotations and ink markup to an existing image. The data flow for PutImageContrastEnhancedMetadata is shown in figure 26.

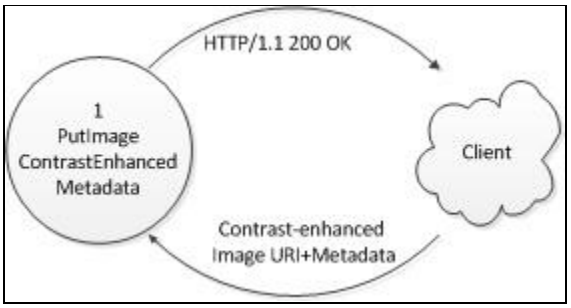


Figure 26. PutImageContrastEnhancedMetadata.

Request Syntax:
PUT /http://{host}:{port}/ARLImageProcessing/images/contrastenhanced/{imageID}

Input Parameters	Definition	Use
imageID	the guid of contrast-enhanced image's metadata to replace	required
URI	the URI of the contrast-enhanced image	copy from current metatadata
MediaType	the mime type of the contrast-enhanced image	copy from current metatadata
Title	the title of the image	optional
DateCreated	the date and time the image was contrast enhanced	copy from current metadata to preserve the orignal creation date or replace with current date/time to preserve the modification timestamp
ParentURI	the URI of parent image which was contrast-enhanced	copy from current metatadata
ContrastEnhancementParameters .PercentageLow	lower tail of the accumulated histogram	copy from current metatadata
ContrastEnhancementParameters .PercentageHigh	upper tail of the accumulated histogram	copy from current metatadata
Annotations	textual and graphical annotations	0..1
Annotations.Text	any textual annotations for the image	optional
Annotations.Text.X	X pixel position for the start of the text	required
Annotations.Text.Y	Y pixel position for the start of the text	required
Annotations.Text.Font	optional font the text	optional
Annotations.Text.Size	optional size for the font	optional
Annotations.Text.Color	optional ARGB color for the text	optional
Annotations.ink	any InkML markup for the image	optional

```
PUT /http://{host}:{port}/ARLImageProcessing/images/contrastenhanced/07659a20-6ac3-42bf-a6a4-93dc26527a39
HTTP/1.1

<?xml version="1.0" encoding="utf-8" ?>
<Metadata URI="http://mule01.lsn.arl:8080/ARLImageProcessing/contrastenhanced/07659a20-6ac3-42bf-a6a4-
93dc26527a39">
   <MediaType>image/png</MediaType>
   <Title>Contrast Enhanced Face</Title>
   <DateCreated>2012-08-29T14:18:45.9298657-04:00</DateCreated>
   <ParentURI>http://mule01.lsn.arl:8080/ARLImageProcessing/images/9618d68b-0f46-4f28-8ef8-
7dd4765d64ad</ParentURI>
   <ContrastEnhancementParameters PercentageLow="0.5" PercentageHigh="0.002" />
   <Annotations>
      <Text X="191" Y="186" Font="Segoe360" Size="12" Color="#FFFF0000">Some Text</Text>
      <Text X="227" Y="315" Font="Segoe360" Size="12" Color="#FFFF0000">Some More Text</Text>
      <ink>
         <definitions>
            <brush xml:id="1">
               <brushProperty name="width" value="2.00314960629921" />
               <brushProperty name="height" value="2.00314960629921" />
               <brushProperty name="color" value="#FF00FFFF" />
               <brushProperty name="tip" value="ellipse" />
            </brush>
         </definitions>
         <trace brushRef="1">38.5566032170582 192.222980271295, 39.4092523017537 191.65161538904,
39.4069967680897 190.702712261337</trace>
      </ink>
   </Annotations>
</Metadata>
```

Listing 53. Example PutImageContrastEnhancedMetadata request.

```
HTTP/1.1 200 OK
```

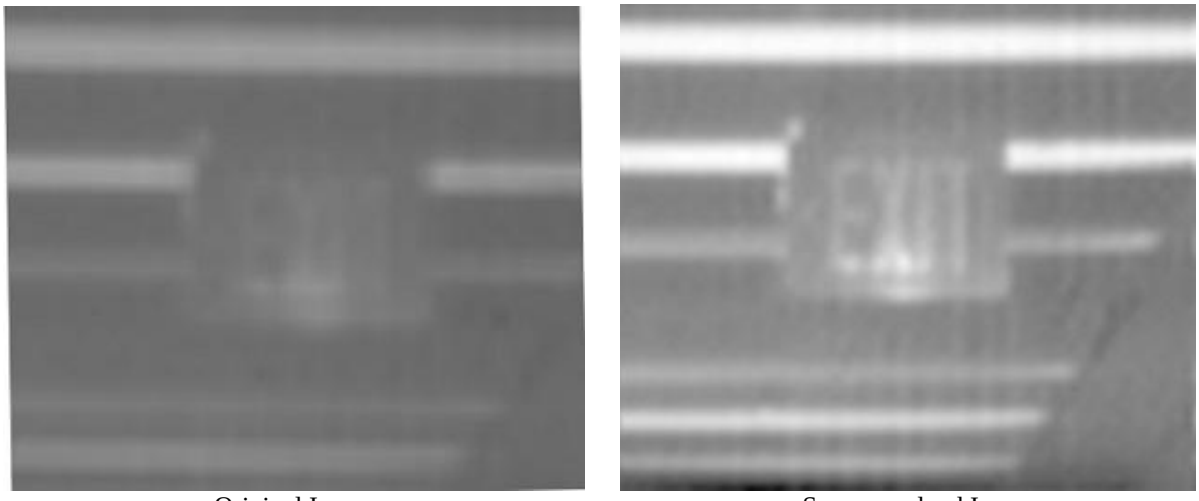
Listing 54. Example PutImageContrastEnhancedMetadata response.

7. Image Super Resolution Services

In this section, we describe the Web services available for super-resolving an image. Super-resolution algorithms process multiple images to produce an image with higher resolution than the source ones. The algorithm used here was developed in the ARL's SEDD Image Processing Branch (Young and Driggers 2006). The following abstract sums up the algorithm and its use:

"We present a superresolution image reconstruction from a sequence of aliased imagery. The subpixel shifts (displacement) among the images are unknown due to the uncontrolled natural jitter of the imager. A correlation method is utilized to estimate subpixel shifts between each low-resolution aliased image with respect to a reference image. An error-energy reduction algorithm is derived to reconstruct the high-resolution alias-free output image. The main feature of this proposed error-energy reduction algorithm is that we treat the spatial samples from low-resolution images that possess unknown and irregular (uncontrolled) subpixel shifts as a set of constraints to populate an oversampled (sampled above the desired output bandwidth) processing array. The estimated subpixel locations of these samples and their values constitute a spatial domain constraint. Furthermore, the bandwidth of the alias-free image (or the sensor imposed bandwidth) is the criterion used as a spatial frequency domain constraint on the oversampled processing array." (Young and Driggers 2006)

An example of a super-resolved image is shown in image 4.



Original Image Super-resolved Image

Image 4. Super0-resolution.

7.1 PostCollectionSuperResolution

Perform a super-resolution of the images contained in the specified by the ParentURI and return the server-generated URI of the super-resolved image. The typical steps involved with super-resolving an image are to first post a sequence of images of the same subject from the same camera angle, preserving the server-generated URIs for these images as they are uploaded to the server; post a new collection containing these URIs; and finally to post the super-resolution parameters. If the base images and collection already exist then the first 2 steps can be skipped. The data flow for PostCollectionSuperResolution is shown in figure 27.

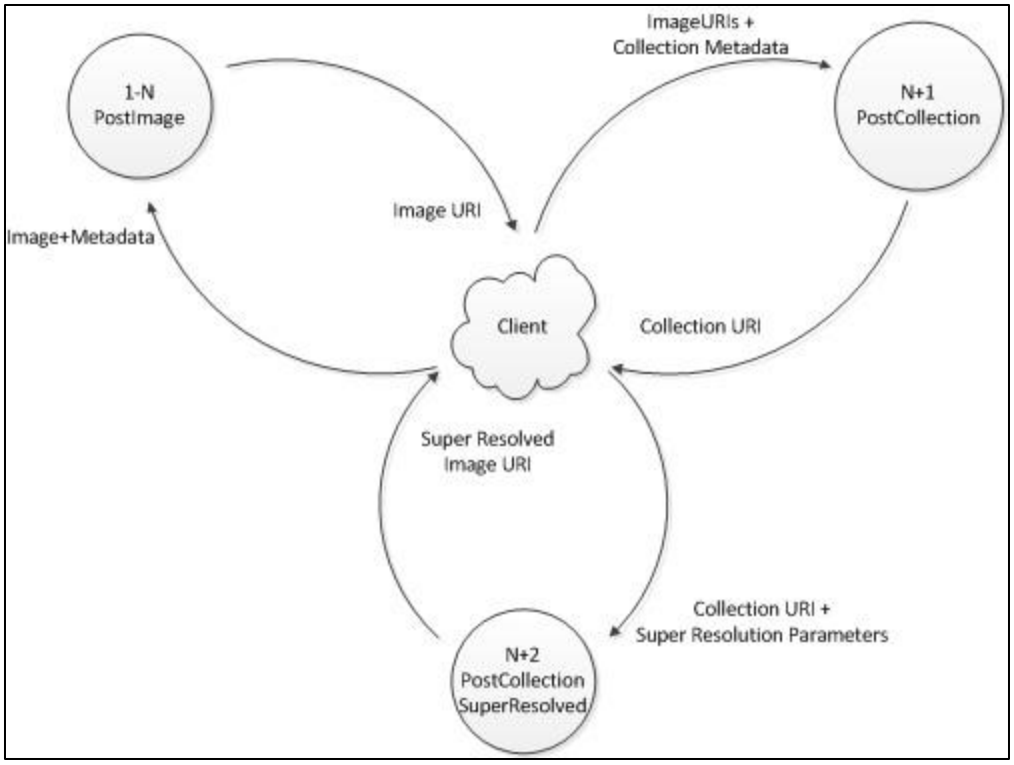


Figure 27. PostCollectionSuperResolution.

The super-resolution image reconstruction algorithm uses a sequence of low resolution undersampled frames containing sub-pixel shift to generate a super-resolved image recovering high frequency information. Four parameters are used by the ARL super-resolution algorithm: *quality*, *speed*, *inset*, and *align*. *quality* (alias for *nr*) is the upsampling factor for the reconstruction grid—it is an even integer and can be typically set to 1.5 times the number of frames. However, increasing *quality* lengthens runtime—for quick demonstrations a setting of 10 is appropriate. *speed* (alias for *per*) controls the frequency bandwidth of the super-resolved image—it is a decimal between 0 and 1, and can be typically set to 0.7. Usually *per* can be increased if the number of frames is small (< 9 frames). The *inset* variable is an integer pixel inset variable that defines a sub-region centered within the user-defined region for frame registration purposes. A setting of 5 for *inset* will work in most scenarios. The last variable *align* is used to indicate whether the server should align the images or whether they've already been manually aligned.

Request Syntax:
POST http://{host}:{port}/ARLImageProcessing/collections/superresolved

Input Parameters	Definition	Use
Title	the title of the image	optional
ParentURI	the URI of the collection containing the images to super resolve	required
Chip.LowerRightX	the X pixel coordinate of the lower right corner of the chip to extract from the parent images	required
Chip.LowerRightY	the Y pixel coordinate of the lower right corner of the chip to extract from the parent images	required
Chip.UpperLeftX	the X pixel coordinate of the upper left corner of the chip to extract from the parent	required
Chip.UpperLeftY	the Y pixel coordinate of the upper left corner of the chip to extract from the parent	required
SuperResolutionParameters.Align	should the base images be aligned?	required
SuperResolutionParameters.Inset	defines a sub-region within the Chip for frame registration purposes	required
SuperResolutionParameters.Quality	upsampling factor for the reconstruction grid	required
SuperResolutionParameters.Speed	frequency bandwidth of the super-resolved image	required

Output Parameters	Definition	Cardinality
string	the server-generated URI of the super-resolved image	1..1

```
POST /http://{host}:{port}/ARLImageProcessing/collections/superresolved HTTP/1.1

<?xml version="1.0" encoding="utf-8"?>
<Metadata xmlns:inkml="http://www.w3.org/2003/InkML"
xmlns="http://arl.army.mil/http://{host}:{port}/ARLImageProcessing/Metadata.xsd">
   <Title>Super-resolved Face</Title>
   <ParentURI>http://mule01.lsn.arl:8080/ARLImageProcessing/collections/45695db1-9605-438e-a7e5-
d8eb2d4c75b1</ParentURI>
   <Chip LowerRightX="119" LowerRightY="70" UpperLeftX="72" UpperLeftY="29" />
   <SuperResolutionParameters Align="true" Inset="5" Quality="10" Speed="0.55" />
</Metadata>
```

Listing 55. Example PostCollectionSuperResolution request.

```
HTTP/1.1 201 Created

<?xml version="1.0" encoding="utf-8"?>
<string>http://mule01.lsn.arl:8080/ARLImageProcessing/collections/superresolved/d35c6f84-9cfc-4b82-8b1b-
cf11d8a02641</string>
```

Listing 56. Example PostCollectionSuperResolution response.

7.2 GetCollectionSuperResolved

Retrieve a super-resolved image and its metadata. The data flow for GetCollectionSuperResolved is shown in figure 28.

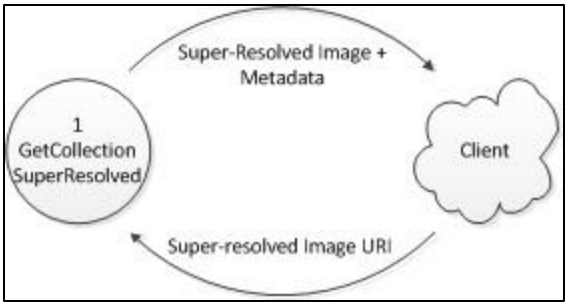


Figure 28. GetCollectionSuperResolved.

Request Syntax:
GET http://{host}:{port}/ARLImageProcessing/collections/superresolved/{imageID}

Input Parameters	Definition	Use
imageID	the guid of the super-resolved image to retrieve	required

Output Parameters	Definition	Cardinality
URI	the URI of the super-resolved image	1..1
MediaType	"image/png"	0..1
Title	the title of the image	0..1
DateCreated	the date and time the image was super resolved	
ParentURI	the URI of the collection containing the images used to super resolve this image	1..1
Chip	the coordinates of the chip extracted from each to use as the basis for super resolution	1..1
Chip.LowerRightX	the X pixel coordinate of the lower right corner of the chip extracted from the parent images which were super resolved	1..1
Chip.LowerRightY	the Y pixel coordinate of the lower right corner of the chip extracted from the parent images which were super resolved	1..1
Chip.UpperLeftX	the X pixel coordinate of the upper left corner of the chip extracted from the parent images which were super resolved	1..1
Chip.UpperLeftY	the Y pixel coordinate of the upper left corner of the chip extracted from the parent images which were super resolved	1..1
SuperResolution	the parameters used to super reolve this image	1..1
SuperResolutionParameters.Align	were base images aligned using the inset parameter?	1..1
SuperResolutionParameters.Inset	defines a sub-region with the Chip for frame registration purposes	1..1
SuperResolutionParameters.Quality	upsampling factor the reconstruction grid	1..1
SuperResolutionParameters.Speed	frequency bandwidth of the super-resolved image	1..1
Annotations	textual and graphical annotations	0..1
Annotations.Text	any textual annotations for the image	0..N
Annotations.Text.X	X pixel position for the start of the text	1..1

Annotations.Text.Y	Y pixel position for the start of the text	1..1
Annotations.Text.Font	optional font the text	0..1
Annotations.Text.Size	optional size for the font	0..1
Annotations.Text.Color	optional ARGB color for the text	0..1
Annotations.ink	any InkML markup for the image	0..1
Media	the base 64 ASCII encoding of the image	1..1

```
GET /http://{host}:{port}/ARLImageProcessing/collections/superresolved/09e13675-e4ec-4676-940e-7a7833cff61b
HTTP/1.1
```

Listing 57. Example GetCollectionSuperResolved request.

```
HTTP/1.1 200 OK

<?xml version="1.0" encoding="utf-8" ?>
<Metadata URI="http://mule01.lsn.arl:8080/ARLImageProcessing/collections/superresolved/09e13675-e4ec-4676-940e-7a7833cff61b">
   <MediaType>image/png</MediaType>
   <Title>Super Resolved Face</Title>
   <DateCreated>2012-08-29T14:21:27.9673503-04:00</DateCreated>
   <ParentURI>http://mule01.lsn.arl:8080/ARLImageProcessing/collections/16e63de6-dff5-4118-b5eb-198bce211ed2</ParentURI>
   <Chip LowerRightX="370" LowerRightY="275" UpperLeftX="283" UpperLeftY="176" />
   <SuperResolutionParameters Align="true" Inset="5" Quality="10" Speed="0.55" />
   <Media>iVBORw0KGg ...</Media>
</Metadata>
```

Listing 58. Example GetCollectionSuperResolved response.

7.3 GetCollectionSuperResolvedMetadata

Retrieve a super-resolved image's metadata. The data flow for GetCollectionSuperResolved is shown in figure 29.

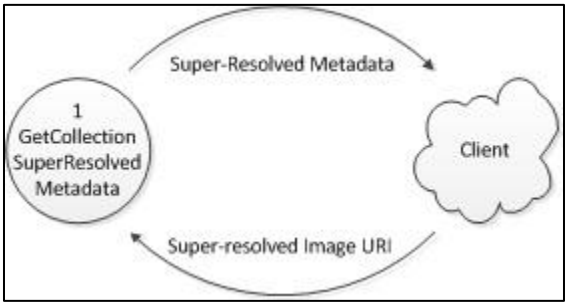


Figure 29. GetCollectionSuperResolvedMetadata

Request Syntax:
GET http://{host}:{port}/ARLImageProcessing/collections/superresolved/{imageID}/metadata

Input Parameters	Definition	Use
imageID	the guid of the super-resolved image metadata to retrieve	required

Output Parameters	Definition	Cardinality
URI	the URI of the super-resolved image	1..1
MediaType	"image/png"	0..1
Title	the title of the image	0..1
DateCreated	the date and time the image was super resolved	
ParentURI	the URI of the collection containing the images used to super resolve this image	1..1
Chip	the coordinates of the chip to extract from each to use as the basis fro super resolution	1..1
Chip.LowerRightX	the X pixel coordinate of the lower right corner of the chip extracted from the parent images which were super resolved	1..1
Chip.LowerRightY	the Y pixel coordinate of the lower right corner of the chip extracted from the parent images which were super resolved	1..1
Chip.UpperLeftX	the X pixel coordinate of the upper left corner of the chip extracted from the parent images which were super resolved	1..1
Chip.UpperLeftY	the Y pixel coordinate of the upper left corner of the chip extracted from the parent images which were super resolved	1..1
SuperResolution	the parameters used to super reolve this image	1..1
SuperResolutionParameters.Align	were base images aligned using the inset parameter?	1..1
SuperResolutionParameters.Inset	defines a sub-region with the Chip for frame registration purposes	1..1
SuperResolutionParameters.Quality	upsampling factor the reconstruction grid	1..1
SuperResolutionParameters.Speed	frequency bandwidth of the super-resolved image	1..1
Annotations	textual and graphical annotations	0..1
Annotations.Text	any textual annotations for the image	0..N
Annotations.Text.X	X pixel position for the start of the text	1..1
Annotations.Text.Y	Y pixel position for the start of the text	1..1
Annotations.Text.Font	optional font the text	0..1
Annotations.Text.Size	optional size for the font	0..1
Annotations.Text.Color	optional ARGB color for the text	0..1
Annotations.ink	any InkML markup for the image	0..1

```
GET http://{host}:{port}/ARLImageProcessing/collections/superresolved/07659a20-6ac3-42bf-a6a4-93dc26527a39/metadata HTTP/1.1
```

Listing 59. Example GetCollectionSuperResolvedMetadata request.

```
HTTP/1.1 200 OK

<?xml version="1.0" encoding="utf-8" ?>
<Metadata URI="http://mule01.lsn.arl:8080/ARLImageProcessing/collections/superresolved/07659a20-6ac3-42bf-a6a4-93dc26527a39">
   <MediaType>image/png</MediaType>
   <Title>Super Resolved Face</Title>
   <DateCreated>2012-08-29T14:18:45.9298657-04:00</DateCreated>
   <ParentURI>http://mule01.lsn.arl:8080/ARLImageProcessing/collections/9618d68b-0f46-4f28-8ef8-7dd4765d64ad</ParentURI>
   <Chip LowerRightX="252" LowerRightY="376" UpperLeftX="152" UpperLeftY="147" />
   <SuperResolutionParameters Align="true" Inset="5" Quality="10" Speed="0.55" />
</Metadata>
```

Listing 60. Example GetCollectionSuperResolvedMetadata response.

7.4 GetCollectionSuperResolvedOverlay

Retrieve a transparent image consisting of just the textual and graphical annotations of a super-resolved image. Applications that do not want to render the annotations and markup themselves can call this service to retrieve a transparent image of just the markup for a super-resolved image and then overlay it on top of the image to see the annotations. The data flow for GetCollectionSuperResolvedOverlay is shown in figure 30.

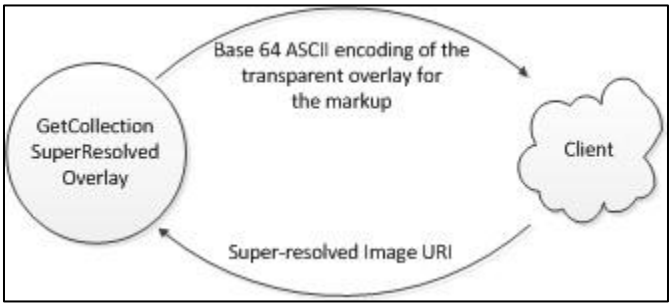


Figure 30. GetCollectionSuperResolvedOverlay.

Request Syntax:
GET http://{host}:{port}/ARLImageProcessing/collections/superresolved/{imageID}/overlay

Input Parameters	Definition	Use
imageID	the guid of the annotated super-resolved image	required

Output Parameters	Definition	Cardinality
base64Binary	the base 64 ASCII encoding of the overlay image	1..1

```
GET http://{host}:{port}/ARLImageProcessing/collections/superresolved/c402ba85-1a7b-47b5-9337-9cd8043e8c6b/metadata
```

Listing 61. Example GetCollectionSuperResolvedOverlay request.

```
HTTP/1.1 200 OK

<?xml version="1.0" encoding="utf-8" ?>
<base64Binary>iVBORw0KGgoAAAANSUhE ...</base64Binary>
```

Listing 62. Example GetCollectionSuperResolvedOverlay response.

7.5 GetCollectionSuperResolvedList

Retrieve the metadata associated with all of the super-resolved images. The data flow for GetCollectionSuperResolvedList is shown in figure 31.

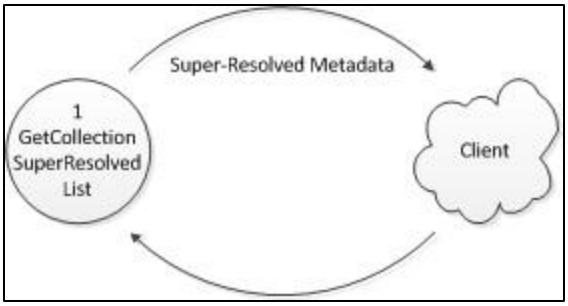


Figure 31. GetCollectionSuperResolvedList.

Request Syntax:
GET http://{host}:{port}/ARLImageProcessing/collections/superresolved

Output Parameters	Definition	Cardinality
Metadata	the metadata for a single super reolved image	0..N
Metadata.URI	the URI of the super-resolved image	1..1
Metadata.MediaType	"image/png"	0..1
Metadata.Title	the title of the image	0..1
Metadata.DateCreated	the date and time the image was super resolved	
ParentURI	the URI of the collection containing the images used to super resolve this image	1..1
Metadata.Chip	the coordinates of the chip to extract from each to use as the basis fro super resolution	1..1
Metadata.Chip.LowerRightX	the X pixel coordinate of the lower right corner of the chip extracted from the parent images which were super resolved	1..1
Metadata.Chip.LowerRightY	the Y pixel coordinate of the lower right corner of the chip extracted from the parent images which were super resolved	1..1
Metadata.Chip.UpperLeftX	the X pixel coordinate of the upper left corner of the chip extracted from the parent images which were super resolved	1..1
Metadata.Chip.UpperLeftY	the Y pixel coordinate of the upper left corner of the chip extracted from the parent images which were super resolved	1..1
Metadata.SuperResolution	the parameters used to super reolve this image	1..1
Metadata.SuperResolutionParameters.Align	were base images aligned using the inset parameter?	1..1
Metadata.SuperResolutionParameters.Inset	defines a sub-region with the Chip for frame registration purposes	1..1
Metadata.SuperResolutionParameters.Quality	upsampling factor the reconstruction grid	1..1
Metadata.SuperResolutionParameters.Speed	frequency bandwidth of the super-resolved image	1..1
Metadata.Annotations	textual and graphical annotations	0..1
Metadata.Annotations.Text	any textual annotations for the image	0..N
Metadata.Annotations.Text.X	X pixel position for the start of the text	1..1
Metadata.Annotations.Text.Y	Y pixel position for the start of the text	1..1
Metadata.Annotations.Text.Font	optional font the text	0..1

Metadata.Annotations.Text.Size	optional size for the font	0..1
Metadata.Annotations.Text.Color	optional ARGB color for the text	0..1
Metadata.Annotations.ink	any InkML markup for the image	0..1

```
GET /http://{host}:{port}/ARLImageProcessing/collections/superresolved HTTP/1.1
```

Listing 63. Example GetCollectionSuperResolvedList request.

```
HTTP/1.1 200 OK

<?xml version="1.0" encoding="utf-8" ?>
<ArrayOfMetadata>
  <Metadata URI="http://mule01.lsn.arl:8080/ARLImageProcessing/collections/superresolved/07659a20-6ac3-42bf-a6a4-93dc26527a39">
    <MediaType>image/png</MediaType>
    <Title>Super Resolved License Tag</Title>
    <DateCreated>2012-08-29T14:18:45.9298657-04:00</DateCreated>
    <ParentURI>http://mule01.lsn.arl:8080/ARLImageProcessing/collections/9618d68b-0f46-4f28-8ef8-7dd4765d64ad</ParentURI>
    <Chip LowerRightX="252" LowerRightY="376" UpperLeftX="152" UpperLeftY="147"/>
    <SuperResolutionParameters Align="true" Inset="5" Quality="10" Speed="0.55"/>
  </Metadata>
  <Metadata URI="http://mule01.lsn.arl:8080/ARLImageProcessing/collections/superresolved/09e13675-e4ec-4676-940e-7a7833cff61b">
    <MediaType>image/png</MediaType>
    <Title>Super Resolved Face</Title>
    <DateCreated>2012-08-29T14:21:27.9673503-04:00</DateCreated>
    <ParentURI>http://mule01.lsn.arl:8080/ARLImageProcessing/collections/16e63de6-dff5-4118-b5eb-198bce211ed2</ParentURI>
    <Chip LowerRightX="370" LowerRightY="275" UpperLeftX="283" UpperLeftY="176"/>
    <SuperResolutionParameters Align="true" Inset="5" Quality="10" Speed="0.55"/>
  </Metadata>
</ArrayOfMetadata>
```

Listing 64. Example GetCollectionSuperResolvedList response.

7.6 DeleteCollectionSuperResolved

Delete a super-resolved image. Note that any references to the image in any collections will be left dangling. The data flow for DeleteCollectionSuperResolved is shown in figure 32.

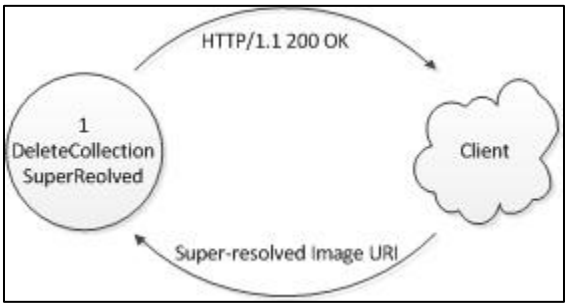


Figure 32. DeleteCollectionSuperResolved.

Request Syntax:
DELETE /http://{host}:{port}/ARLImageProcessing/collections/superresolved/{imageID}

Input Parameters	Definition	Use
imageID	the guid of the super-resolved image to delete	required

```
DELETE /http://{host}:{port}/ARLImageProcessing/collections/superresolved/c734551e-bc64-453b-a361-4dbe17f1f356 HTTP/1.1
```

Listing 65. Example DeleteCollectionSuperResolved request.

```
HTTP/1.1 200 OK
```

Listing 66. Example DeleteCollectionSuperResolved response.

7.7 PutCollectionSuperResolvedMetadata

Replace a super-resolved image's metadata. Note that the image's metadata is entirely replaced and not merged or otherwise checked in any manner. Use cautiously since even the image's URI can be changed or deleted with this method, resulting in not only a corrupt image, but also corrupting any collections it belongs to. To add to the metadata, the current metadata should first be retrieved and copied into the new metadata. This method is typically used to add textual annotations and ink markup to an existing image. The data flow for PutCollectionSuperResolvedMetadata is shown in figure 33.

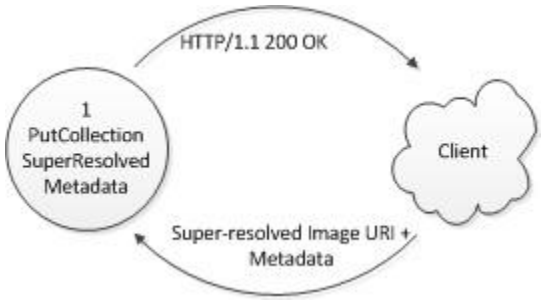


Figure 33. PutCollectionSuperResolvedMetadata.

Request Syntax:
PUT /http://{host}:{port}/ARLImageProcessing/collections/superresolved/{imageID}

Input Parameters	Definition	Use
imageID	the guid of super-resolved image's metadata to replace	required
URI	the URI of the super-resolved image	copy from current metadata
MediaType	the mime type of the super-resolved image	copy from current metatadata
Title	the title of the image	optional
DateCreated	the date and time the image was super resolved	copy from current metadata to preserve the orignal creation date or replace with current date/time to preserve the modification timestamp
ParentURI	the URI of the collection containing the images used to super resolve this image	copy from current metatadata
Chip.LowerRightX	the X pixel coordinate of the lower right corner of the chip extracted from the parent images which were super resolved	copy from current metatadata
Chip.LowerRightY	the Y pixel coordinate of the lower right corner of the chip extracted from the parent images which were super resolved	copy from current metatadata
Chip.UpperLeftX	the X pixel coordinate of the upper left corner of the chip extracted from the parent images which were super resolved	copy from current metatadata
Chip.UpperLeftY	the Y pixel coordinate of the upper left corner of the chip extracted from the parent images which were super resolved	copy from current metatadata
SuperResolutionParameters.Align	were the base images aligned	copy from current metatadata
SuperResolutionParameters.Inset	defines a sub-region with the Chip for frame registration purposes	copy from current metatadata
SuperResolutionParameters.Quality	upsampling factor the reconstruction grid	copy from current metatadata
SuperResolutionParameters.Speed	frequency bandwidth of the super-resolved image	copy from current metatadata
Annotations	textual and graphical annotations	0..1
Annotations.Text	any textual annotations for the image	optional
Annotations.Text.X	X pixel position for the start of the text	required
Annotations.Text.Y	Y pixel position for the start of the text	required
Annotations.Text.Font	optional font the text	optional
Annotations.Text.Size	optional size for the font	optional
Annotations.Text.Color	optional ARGB color for the text	optional
Annotations.ink	any InkML markup for the image	optional

```
PUT /http://{host}:{port}/ARLImageProcessing/collections/superresolved/07659a20-6ac3-42bf-a6a4-
93dc26527a39 HTTP/1.1

<?xml version="1.0" encoding="utf-8" ?>
<Metadata URI="http://mule01.lsn.arl:8080/ARLImageProcessing/collections/superresolved/07659a20-6ac3-42bf-
a6a4-93dc26527a39">
   <MediaType>image/png</MediaType>
   <Title>Super Resolved Face</Title>
   <DateCreated>2012-08-29T14:18:45.9298657-04:00</DateCreated>
   <ParentURI>http://mule01.lsn.arl:8080/ARLImageProcessing/collections/9618d68b-0f46-4f28-8ef8-
7dd4765d64ad</ParentURI>
   <Chip LowerRightX="252" LowerRightY="376" UpperLeftX="152" UpperLeftY="147" />
   <SuperResolutionParameters Align="true" Inset="5" Quality="10" Speed="0.55" />
   <Annotations>
      <Text X="191" Y="186" Font="Segoe360" Size="12" Color="#FFFF0000">Some Text</Text>
      <Text X="227" Y="315" Font="Segoe360" Size="12" Color="#FFFF0000">Some More Text</Text>
      <ink>
         <definitions>
            <brush xml:id="1">
               <brushProperty name="width" value="2.00314960629921" />
               <brushProperty name="height" value="2.00314960629921" />
               <brushProperty name="color" value="#FF00FFFF" />
               <brushProperty name="tip" value="ellipse" />
            </brush>
         </definitions>
         <trace brushRef="1">38.5566032170582 192.222980271295, 39.4092523017537 191.65161538904,
39.4069967680897 190.702712261337</trace>
      </ink>
   </Annotations>
</Metadata>
```

Listing 67. Example PutCollectionSuperResolvedMetadata request.

```
HTTP/1.1 200 OK
```

Listing 68. Example PutCollectionSuperResolvedMetadata response.

8. Conclusion

In this report, we have presented an architecture based on REST Web services for image processing algorithms. Relying on just two types of resources, images and collections, the architecture accommodates the chaining together of these algorithms to support multiple transformations of an image. Support for both graphical and textual annotation of images is also provided. The architecture is readily extensible to include any number of additional image processing algorithms

9. References

Fielding, R. T. *Architectural Styles and the Design of Network-based Software Architectures.* Doctoral Dissertation, University of California, Irvine, 2000.

Maschel, R.; Young, S. S. Locally Adaptive Contrast Enhancement and Dynamic Range Compression. *Proceedings of SPIE Vol. 8355.* SPIE, 2012.

W3C. *Ink Markup Language (InkML).* September 20, 2011. http://www.w3.org/TR/InkML/ (accessed October 23, 2012).

Young, S. S.; Driggers, R. G. Superresolution Image Reconstruction from a Sequence of Aliased Imagery. *Applied Optics* **2006**, *45* (21), 5073–5085.

Young, S. S.; Driggers, R. G.; Teaney, B. P.; Jacobs, E. L. Adaptive Deblurring of Noisy Images. *Applied Optics* **2007**, *46* (5), 744–752.

1 (PDF only)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
1 HC	DIRECTOR US ARMY RESEARCH LAB IMAL HRA
1 PDF	DIRECTOR US ARMY RESEARCH LAB RDRL CIO LL
2 PDF	DIRECTOR US ARMY RESEARCH LAB RDRL-SES-E S YOUNG S HU
5 PDF	DIRECTOR US ARMY RESEARCH LAB RDRL-CII-B R WINKLER C SCHLESIGER RDRL-CII-C M THOMAS M MITTRICK J RICHARDSON

INTENTIONALLY LEFT BLANK.